

# RELASJONSALGEBRA

## Regning med relasjoner

# Relasjonsalgebraen

- definerer en mengde av operasjoner på relasjoner
- gir oss et språk til å beskrive spørsmål om innholdet i relasjonene
- er et **prosedyralt** spørrespråk:  
Vi sier *hvordan* svaret skal beregnes  
(Alternativet er **deklarative** spørrespråk som SQL hvor vi bare sier *hva svaret skal oppfylle*)
- utgjør det teoretiske grunnlaget for prosessering av SQL-spørringer mot relasjonsdatabaser ('SQL-queries')

# Algebra

- **Domene** (samling av verdier)
- **Atomære operander**
  - Konstanter  
(representerer konkrete verdier i domenet)
  - Variable  
(representerer vilkårlige verdier fra domenet)
- **Operatorer**
  - Tar som argumenter operander
  - Leverer som resultat en operand
- **Uttrykk**
  - Bygges av atomære operander med operatorer og parenteser

# Eksempel: Heltallsalgebra

- Domene: Heltallene
- Konstanter: ..., -3, -2, -1, 0, 1, 2, 3, ...
- Variable: x, y, z, ...
- Operatorer: +, -, ×, /
- Eksempler på uttrykk:

$$2 + 5$$

$$((2-x) \times 5) + (y/z)$$

# Klassisk relasjonsalgebra

- Domene: Endelige relasjoner
- Atomære operander:
  - Konstanter: Alle endelige relasjoner
  - Variable: Representerer vilkårlige endelige relasjoner
- Operatorer:
  - union
  - snitt
  - differanse
  - kartesisk produkt
  - projeksjon
  - seleksjon
  - join
  - renavning
  - divisjon

# Mengdeoperatorer

- **Union:**  $R \cup S$
- **Snitt:**  $R \cap S$
- **Differanse:**  $R - S$

R og S må ha skjemaer med identiske attributtmengder (og identiske domener)

Før operasjonen utføres, må S ordnes slik at attributtene kommer i samme rekkefølge som i R

# Union

- $R \cup S$  er en relasjon hvor
  - alle tupler som er i  $R$  eller i  $S$  eller i både  $R$  og  $S$ , er i  $R \cup S$   
(Om  $t$  er i både  $R$  og  $S$ , er  $t$  likevel bare representert én gang i  $R \cup S$   
(fordi en relasjon er en *mengde*))
  - ingen andre tupler forekommer i  $R \cup S$

Eksempel på vanlig mengdeunion:

$$\{a,b,c\} \cup \{a,c,d\} = \{a,b,c,d\}$$

# Snitt

- $R \cap S$  er en relasjon hvor
  - alle tupler som er i både R og S, er i  $R \cap S$
  - ingen andre tupler forekommer i  $R \cap S$

Eksempel på vanlig mengdesnitt:

$$\{a,b,c\} \cap \{a,c,d\} = \{a,c\}$$



# Differanse

- R–S er en relasjon hvor
  - alle tupler som er i R, men ikke i S, er i R–S
  - ingen andre tupler forekommer i R–S

Eksempel på vanlig mengdedifferanse:

$$\{a,b,c\} - \{a,c,d\} = \{b\}$$

# Operatorer som fjerner deler av en relasjon

- Seleksjon:  $\sigma_C(R)$
- Projeksjon:  $\pi_L(R)$

# Seleksjon

- $\sigma_C(R)$  er relasjonen som fås fra  $R$  ved å velge ut de tuplene i  $R$  som tilfredsstillers betingelsen  $C$
- $C$  er et vilkårlig boolsk uttrykk bygget opp fra atomer på formen  $op_1 \varphi op_2$  der
  - operandene  $op_1$  og  $op_2$  er
    - enten to attributter i  $R$  med samme domene
    - eller ett attributt i  $R$  og en konstant fra dette attributtets domene
  - operatoren  $\varphi \in \{ =, \neq, <, >, \leq, \geq, \text{LIKE} \}$

# Projeksjon

- $\pi_L(R)$  hvor  $R$  er en relasjon og  $L$  er en liste av attributter i  $R$ , er relasjonen som fås fra  $R$  ved å velge ut kolonnene til attributtene i  $L$ 
  - Relasjonen har et skjema med attributtene i  $L$
  - Ingen tupler skal forekomme flere ganger i  $\pi_L(R)$

# Renavning

- $\rho_{S(A_1, A_2, \dots, A_n)}(R)$  renavner  $R$  til en relasjon med navn  $S$  og attributter  $A_1, A_2, \dots, A_n$
- $\rho_S(R)$  renavner  $R$  til en relasjon med navn  $S$   
Attributtnavnene fra  $R$  beholdes

# Operatorer som spleiser tupler

- Kartesisk produkt:  $R \times S$
- Naturlig join:  $R \bowtie S$
- Theta-join:  $R \bowtie_{\theta} S$

# Kartesisk produkt

- $R \times S$  er relasjonen som fås fra  $R$  og  $S$  ved å danne alle mulige sammensetninger av ett tuppel fra  $R$  og ett tuppel fra  $S$
- Vi sier ofte at et tuppel  $t$  fra  $R$  og et tuppel  $u$  fra  $S$  blir **konkateneret** til et tuppel  $v=tu$  i  $R \times S$
- I resultatskjemaet løses eventuell navnelikhet mellom attributter i  $R$  og  $S$  ved å **kvalifisere** navnene med opprinnelsesrelasjonen:  $R.A$ ,  $S.A$
- Hvis  $R$  og  $S$  er samme relasjon, må en av dem først renavnes

# Naturlig join

- $R \bowtie S$  er relasjonen som fås fra  $R$  og  $S$  ved å danne alle mulige sammensmeltinger av ett tuppel fra  $R$  med ett fra  $S$  der tuplene skal stemme overens i samtlige attributter med sammenfallende navn
  - Fellesattributtene forekommer bare en gang i de sammensmeltede attributtene
  - Resultatskjemaet har attributtene i  $R$  etterfulgt av de attributtene i  $S$  som ikke også forekommer i  $R$



# Hengetupler

- Et **hengetuppel** (dangling tuple) er et tuppel i en av relasjonene som ikke har noe matchende tuppel i den andre relasjonen
- Hengetupler får ingen representant i resultatrelasjonen etter en join

# Theta-join

- Generalisering av naturlig join
- Relasjonen  $R \bowtie_{\theta} S$ , hvor  $\theta$  er en betingelse (boolsk uttrykk), fremkommer slik:
  1. Beregn  $R \times S$
  2. Velg ut de tuplene som tilfredsstill betingelsen  $\theta$
- Atomene i  $\theta$  har formen  $A \varphi B$  der  $A$  og  $B$  er attributter i henholdsvis  $R$  og  $S$ ,  $A$  og  $B$  har samme domene, og  $\varphi \in \{ =, \neq, <, >, \leq, \geq \}$

# Ekvijoin

- Spesialtilfelle av en theta-join  $R \bowtie_{\theta} S$  hvor betingelsen  $\theta$  tilfredstiller følgende krav:
  1.  $\theta$  inneholder ingen andre boolske operatorer enn AND, dvs at  $\theta$  har formen  $\theta_1 \text{ AND } \theta_2 \text{ AND } \dots \text{ AND } \theta_m$
  2. Hver  $\theta_k$  for  $1 \leq k \leq m$  er på formen  $A = B$  der  $A$  er et attributt i  $R$  og  $B$  et attributt i  $S$  hvor  $A$  og  $B$  har samme domene

# Divisjon

- Gitt to relasjoner  $R(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$  og  $S(B_1, B_2, \dots, B_m)$ . **R div S** er en relasjon  $Q(A_1, A_2, \dots, A_n)$  som inneholder et tuppel  $t$  hvis og bare hvis det *for hvert eneste tuppel*  $u$  i  $S$  fins et tuppel  $v$  i  $R$  slik at  $\pi_{A_1, A_2, \dots, A_n}(v)=t$  og  $\pi_{B_1, B_2, \dots, B_m}(v)=u$ .

Hvis vi lar  $tu$  betegne sammensetningen av to tupler  $t$  og  $u$  til ett tuppel, skal altså  $t$  være med i svaret når vi for hver eneste  $u$  i  $S$  har at  $tu$  er med i  $R$ .

# Eksempel på når divisjon brukes

**Spisested** viser hva slags mat hvert spisested serverer

**Menykrav** viser hva slags mat hver person vil/kan spise

## Spisested

| Navn | Matkategori      |
|------|------------------|
| A    | kosher           |
| A    | vegetabilsk      |
| B    | uten melk        |
| B    | lactavegetabilsk |
| B    | glutenfri        |
| B    | kosher           |
| C    | glutenfri        |
| C    | lactavegetabilsk |
| C    | kosher           |
| D    | vegetabilsk      |

## Menykrav

| Navn | Krav             |
|------|------------------|
| Per  | lactavegetabilsk |
| Kari | kosher           |
| Ola  | kosher           |
| Tone | glutenfri        |

# Spisested som dekker alles krav til menyen

$\text{AlleKrav}(\text{Matkategori}) \leftarrow \pi_{\text{Krav}}(\text{Menykrav})$

$\text{MuligeSpisesteder} \leftarrow \text{Spisested} \mathbf{div} \text{AlleKrav}$

# Resultat av divisjonen

MuligeSpisesteder ← Spisested **div** AlleKrav

Spisested

| Navn | Matkategori      |
|------|------------------|
| A    | kosher           |
| A    | vegetabilsk      |
| B    | uten melk        |
| B    | lactavegetabilsk |
| B    | glutenfri        |
| B    | kosher           |
| C    | glutenfri        |
| C    | lactavegetabilsk |
| C    | kosher           |
| D    | vegetabilsk      |

AlleKrav

| Matkategori      |
|------------------|
| lactavegetabilsk |
| kosher           |
| glutenfri        |

MuligeSpisesteder

| Navn |
|------|
| B    |
| C    |

# Divisjon uttrykt ved projeksjon, kartesisk produkt og differanse

$$\begin{aligned} R(A,B) \text{ div } S(B) = \\ \pi_A(R) - \\ \pi_A((\pi_A(R) \times S) - R) \end{aligned}$$



# Andre avledbare operatorer

- $R \cap S = R - (R - S)$ 
  - Så vi trenger ikke  $\cap$
- $R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$ 
  - Så vi trenger ikke  $\bowtie_{\theta}$
- $R \bowtie S = \pi_L(\sigma_{\theta}(R \times S))$ 

hvor L er listen av attributtene i R fulgt av de attributtene i S som ikke forekommer i R, og  $\theta$  er  $R.A_1 = S.A_1$  AND ... AND  $R.A_k = S.A_k$  der  $A_1, \dots, A_k$  er alle attributter som forekommer i både R og S

  - Så vi trenger ikke  $\bowtie$

# En minimal mengde operatører

- Operatorene i mengden  $\{\cup, -, \sigma, \pi, \times, \rho\}$  kan ikke uttrykkes ved noen av de andre operatorene i mengden
  - Dette er altså en minimal og uavhengig mengde av operatører
  - Vi ønsker likevel å ha med de øvrige operatorene fordi det fins effektive algoritmer for dem og fordi det ofte er enklere å finne spørsmålsformuleringer når vi har dem

# Bag

- Kommersielle DBMSer benytter Bag og ikke Set (mengde) som grunntype for å realisere relasjoner
  - Set(D):  
Hvert element i D forekommer maksimalt én gang  
Rekkefølgen på elementene er likegyldig  
 $\{a,b,c\} = \{a,c,b\} = \{a,a,b,c\} = \{c,a,b,a\}$
  - Bag(D):  
Hvert element i D kan forekomme mer enn en gang  
Rekkefølgen på elementene er likegyldig  
 $\{a,b,c\} = \{a,c,b\} \neq \{a,a,b,c\} = \{c,a,b,a\}$

# Hvorfor Bag og ikke Set

- Bag gir mer effektive beregninger av union og projeksjon enn Set
- Ved aggregering trenger vi bagfunksjonalitet
- Men: Bag er mer plasskrevende enn Set

# Relasjonsalgebraens operatører anvendt på Bag

- Definisjonene blir litt annerledes
- Ikke alle algebraiske lover som holder for Set holder for Bag

$$\text{Eks: } (R \cup S) - T = (R - T) \cup (S - T)$$

Når vi på de videre foilene skriver **bagrelasjon**, mener vi et relasjonsskjema + en ekstensjon (instans) hvor ekstensjonen er en *bag*, og ikke en mengde

# Bagunion

- La  $R$  og  $S$  være bagrelasjoner
- Hvis  $t$  er et tuppel som forekommer  $n$  ganger i  $R$  og  $m$  ganger i  $S$ , så forekommer  $t$   $n+m$  ganger i bagrelasjonen  $R \cup S$

Eksempel på vanlig bagunion:

$$\{a,a,b,c,c\} \cup \{a,c,c,c,d\} = \{a,a,a,b,c,c,c,c,d\}$$

# Bagsnitt

- La  $R, S$  være bagrelasjoner
- Hvis  $t$  er et tuppel som forekommer  $n$  ganger i  $R$  og  $m$  ganger i  $S$ , så forekommer  $t$   $\min(n, m)$  ganger i bagrelasjonen  $R \cap S$

Eksempel på vanlig bagsnitt:

$$\{a, a, b, c, c\} \cap \{a, c, c, c, d\} = \{a, c, c\}$$

# Bagdifferanse

- La  $R$ ,  $S$  være bagrelasjoner
- Hvis  $t$  er et tuppel som forekommer  $n$  ganger i  $R$  og  $m$  ganger i  $S$ , så forekommer  $t$   $\max(0, n-m)$  ganger i bagrelasjonen  $R-S$

Eksempel på vanlig bagdifferanse:

$$\{a, a, b, c, c\} - \{a, c, c, c, d\} = \{a, b\}$$



# Bagseleksjon

- Hvis  $R$  er en bagrelasjon, er  $\sigma_\theta(R)$  bagrelasjonen som fås fra  $R$  ved å anvende  $\theta$  på hvert enkelt tuppel individuelt og velge ut de tuplene i  $R$  som tilfredsstillter betingelsen  $\theta$

# Bagprosjeksjon

- Hvis  $R$  er en bagrelasjon og  $L$  er en (ikke tom) liste av attributter, er  $\pi_L(R)$  bagrelasjonen som fås fra  $R$  ved å velge ut kolonnene til attributtene i  $L$
- $\pi_L(R)$  har like mange tupler som  $R$

# Kartesisisk produkt av bager

- $R \times S$  er bagrelasjonen som fås fra bagrelasjonene  $R$  og  $S$  ved å danne alle mulige konkateneringer av ett tuppel fra  $R$  og ett tuppel fra  $S$
- Hvis  $R$  har  $n$  tupler og  $S$  har  $m$  tupler, blir det  $nm$  tupler i  $R \times S$

# Theta-join på bager

- Hvis  $R$  og  $S$  er bagrelasjoner, fremkommer bagrelasjonen  $R \bowtie_{\theta} S$ , hvor  $\theta$  er en betingelse, slik:
  - 1) Beregn  $R \times S$  (kartesisk produkt av bager)
  - 2) Velg ut de tuplene som tilfredsstill betingelsen  $\theta$

# Naturlig join på bager

- Hvis  $R$  og  $S$  er bagrelasjoner, er  $R \bowtie S$  bagrelasjonen som fås ved å sammensmelte matchende tupler i  $R$  og  $S$  individuelt

# Tilleggsoperatorer i relasjonsalgebraen

1. Duplikateliminering
2. Aggregeringsoperatorer
3. Gruppering
4. Sortering
5. Utvidet projeksjon
6. Outerjoin

# Duplikateliminasjon

- $\delta(R)$  fjerner flerforekomster av tupler fra bagrelasjonen  $R$
- Resultatet blir en mengde

# Aggregeringsoperatører

- Anvendes på bager av atomære verdier for et attributt A
- Standard aggregeringsoperatører:
  - SUM(A) :  
Summerer alle verdier i kolonnen til A  
Domenet til A må være numeriske verdier
  - AVG(A) :  
Beregner gjennomsnittet av verdiene i kolonnen til A (Kolonnen må ha minst én verdi)  
Domenet til A må være numeriske verdier



# Aggregeringsoperatører (forts.)

- MIN(A), MAX(A):

Plukker ut minste/største verdi i kolonnen til A  
(Kolonnen må ha minst én verdi)

Domenet til A må ha en *ordningsrelasjon*

For numeriske verdier er dette  $<$

For strenger benyttes leksikografisk ordning

- COUNT(A):

Teller antall tupler i relasjonen som har verdi i kolonnen til A

(dvs. at tupler hvor A er nil, ikke telles med)

# Gruppering

- Benyttes når vi ønsker å anvende en aggregeringsoperator på grupper av verdier
- $\gamma_L(R)$ : L er en liste av elementer på en av følgende former:
  - Et attributt A i R  
[A kalles et **grupperingsattributt**]
  - En aggregeringsoperator anvendt på et attributt A i R av formen  $AGG(A) \rightarrow AggRes$  hvor AGG er en aggregeringsoperator og AggRes er et ubrukt attributtnavn  
[A kalles et **aggregeringsattributt**]
- L får ikke inneholde to like elementer

# Resultatrelasjonen etter gruppering

Gitt grupperingen  $\gamma_L(R)$

Resultatrelasjonen konstrueres slik:

1. Partisjoner  $R$  i grupper,  
én gruppe for hver samling av tupler som er like i samtlige grupperingsattributter i  $L$
2. For hver gruppe, produser et tuppel bestående av
  - i. Grupperingsattributtens verdi i gruppen
  - ii. For hvert aggregeringsattributt i  $L$ , aggregeringen over alle tuplene i gruppen

Resultatrelasjonen får like mange attributter som det er elementer i  $L$ , og attributtnavn som angitt av  $L$

# Sortering

- $\tau_L(R)$ , hvor  $R$  er en relasjon og  $L$  en liste av attributter  $A_1, A_2, \dots, A_k$ , leverer som resultat en liste av tupler som er sortert først etter  $A_1$ , deretter etter  $A_2$  internt i hver bunke med like  $A_1$ -verdier, osv.
- De attributtene som ikke er i listen, ordnes vilkårlig
- Resultatet er en *liste*, så operasjonen er bare meningsfylt som en siste, avsluttende operasjon på relasjoner

# Utvidet projeksjon

- $\pi_L(R)$ , klassisk: L er en liste av attributter i R
- $\pi_L(R)$ , utvidet: L er en liste der hvert element kan være
  - i. Ett enkelt attributt i R
  - ii. Et uttrykk  $A \rightarrow B$ , hvor A er et attributt i R og B er et ubrukt attributtnavn  
Renavner A i R til B i resultatrelasjonen
  - iii. Et uttrykk  $E \rightarrow B$ , hvor E er et uttrykk bygget opp fra attributter i R, konstanter, aritmetiske operasjoner og strengoperasjoner, og B er et ubrukt attributtnavn

# Utvidet projeksjon, resultatrelasjon

- Resultatrelasjonen  $\pi_L(R)$  fås fra R som følger:
  - Betrakt hvert tuppel i R for seg
  - Substituer inn tuppelets verdier for attributtnavnene i L og beregn uttrykkene i L
  - Resultatrelasjonen er en bag med like mange attributter som elementer i L, og med navn som angitt i L

# Outerjoin

- Outerjoin benyttes når man ønsker å ta vare på hengetupler (dangling tuples) fra naturlig join
- $R \overset{\circ}{\bowtie} S$ , **outerjoin**: Start med  $R \bowtie S$   
Legg til hengetupler fra R og S, manglende attributtverdier fylles ut med  $\perp$  (nil)
- $R \overset{\circ}{\bowtie}_L S$ , **left outerjoin**:  
Bare hengetupler fra R legges til
- $R \overset{\circ}{\bowtie}_R S$ , **right outerjoin**:  
Bare hengetupler fra S legges til

# Relasjoner og integritetsregler

- Vi kan uttrykke referanseintegriteter, funksjonelle avhengigheter og flerverdiavhengigheter – og også andre klasser av integritetsregler – i relasjonsalgebraen!



# Integritetsregler formulert i relasjonsalgebraen

1. Hvis  $E$  er et uttrykk i relasjonsalgebraen, så er  $E = \emptyset$  en integritetsregel som sier at  $E$  ikke har noen tupler
2. Hvis  $E_1$  og  $E_2$  er uttrykk i relasjonsalgebraen, så er  $E_1 \subseteq E_2$  en integritetsregel som sier at ethvert tuppel i  $E_1$  også skal være i  $E_2$ 
  - Merk at  $E_1 \subseteq E_2$  og  $E_1 - E_2 = \emptyset$  er ekvivalente, og det er  $E = \emptyset$  og  $E \subseteq \emptyset$  også, så det er tilstrekkelig med en av formene over
  - Strengt tatt er ikke  $\emptyset$  et relasjonsalgebrauttrykk. Vi kunne i stedet ha skrevet  $R - R$  (for en vilkårlig relasjon  $R$  med samme skjema som  $E$ )

# Eksempler på integritetsregler

- **Referanseintegritet:** "A er fremmednøkkel til S", hvor B er primærnøkkelen i S:

$$\pi_A(R) \subseteq \pi_B(S)$$

- **FDer:** "A<sub>1</sub> A<sub>2</sub> ... A<sub>n</sub> → B<sub>1</sub> B<sub>2</sub> ... B<sub>m</sub>" i R:

$$\sigma_{\theta}(\rho_{R_1}(R) \times \rho_{R_2}(R)) = \emptyset$$

hvor  $\theta$  er uttrykket

R1.A<sub>1</sub>=R2.A<sub>1</sub> AND ... AND R1.A<sub>n</sub>=R2.A<sub>n</sub> AND  
(R1.B<sub>1</sub>≠R2.B<sub>1</sub> OR ... OR R1.B<sub>m</sub>≠R2.B<sub>m</sub>)

- **Domeneskranker:**

$$\sigma_{A \neq 'F' \text{ AND } A \neq 'M'}(R) = \emptyset$$