



# ifj

## INF1300— Relasjonsalgebra og SQL, mengder og bager.

Lysark for forelesning v. 3.0



UNIVERSITETET  
I OSLO

# Dagens temaer

- ▶ Relasjonsalgebraen
- ▶ Algebra
- ▶ Heltallsalgebra
- ▶ Klassisk relasjonsalgebra
- ▶ Mengdeoperatorer
- ▶ Union
- ▶ Snitt
- ▶ Differanse
- ▶ Seleksjon
- ▶ Prosjeksjon
- ▶ Kartesisk produkt
- ▶ Join
- ▶ Naturlig join
- ▶ Hengetupler
- ▶ Renavning
- ▶ Divisjon
- ▶ Bag
- ▶ Bagunion, -snitt og -differanse
- ▶ Bageleksjon
- ▶ Bagprojeksjon
- ▶ Kartesisk produkt av bager
- ▶ Join på bager
- ▶ Naturlig join på bager
- ▶ Relasjonsalgebraetolkning av select-setningen
- ▶ Relasjonsalgebra (klassisk og bager) uttrykt i SQL
- ▶ Kostbare operasjoner i SQL

## 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 hvor vi sier *hva* svaret skal oppfylle. (SQL er et **deklarativt** språk.))
- ▶ utgjør det teoretiske grunnlaget for prosessering av SQL-spørringer mot relasjonsdatabaser

# 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* (generelle operander)
  - ▶ Bygges av atomære operander med operatorer og parenteser

# Heltallsalgebra

- ▶ Domene:  $\mathbb{Z}$ , heltallene.
- ▶ Konstanter:  $\dots, -3, -2, -1, 0, 1, 2, 3, 4, \dots$
- ▶ Variable:  $z, i, j, k, m, n, x, y, \dots$
- ▶ Operatorer  $+, -, \cdot, /$
- ▶ Eksempler på uttrykk:  
 $2 + 5$   
 $((2 - k) \cdot 5) + (i/j)$

# Klassisk relasjonsalgebra

- ▶ *Domene*: Endelige relasjoner (endelig mengde av tupler)
- ▶ *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:  $R \setminus S$

*Unionkompabilitet:*  $R$  og  $S$  må ha like mange attributter og attributtene må parvis ha identiske domener. Eks:

$\text{Ansatt}(\text{navn}, \text{tlf}, \text{adr}, \text{aid}) \cup \text{Medlem}(\text{mid}, \text{navn}, \text{adr}, \text{tlf})$

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

$\text{Ansatt}(\text{navn}, \text{tlf}, \text{adr}, \text{aid}) \cup \text{Medlem}(\text{navn}, \text{tlf}, \text{adr}, \text{mid})$ .

(Attributter med like navn må ha samme domene, likeså aid og mid).

# Union

- ▶ For at  $R \cup S$  skal være definert, må  $R$  og  $S$  være *unionkompatible*
- ▶  $R \cup S$  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:  
 $\{a, b, c\} \cup \{b, c, d, e\} = \{a, b, c, d, e\}$



# Union

- ▶ Eksempel på union av relasjoner:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Lise	79023365	Vei 238-5
Geir	25364758	Viken 2

## Medlem

navn	adr	tlf
Lise	Bakken 1	37485960
Liv	Lia 3	19203142
Jan	Heia 7	26374859
Geir	Viken 2	25364758

## Ansatt $\cup$ Medlem

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Lise	79023365	Vei 238-5
Geir	25364758	Viken 2
Liv	19203142	Lia 3
Jan	26374859	Heia 7

# Snitt

- ▶ For at  $R \cap S$  skal være definert, må  $R$  og  $S$  være *unionkompatible*
- ▶  $R \cap S$  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:  
 $\{a, b, c\} \cap \{b, c, d, e\} = \{b, c\}$

# Snitt

Eksempel på snitt mellom relasjoner:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Lise	79023365	Vei 238-5
Geir	25364758	Viken 2

## Medlem

navn	adr	tlf
Lise	Bakken 1	37485960
Liv	Lia 3	19203142
Jan	Heia 7	26374859
Geir	Viken 2	25364758

## Ansatt $\cap$ Medlem

navn	tlf	adr
Lise	37485960	Bakken 1
Geir	25364758	Viken 2

# Differanse

- ▶ For at  $R \setminus S$  skal være definert, må  $R$  og  $S$  være *unionkompatible*
- ▶  $R \setminus S$  er en *relasjon* hvor
  - ▶ alle tupler som er i  $R$ , men ikke i  $S$ , er i  $R \setminus S$
  - ▶ ingen andre tupler forekommer i  $R \setminus S$
- ▶ Eksempel på vanlig mengdedifferanse:  
 $\{a, b, c\} \setminus \{b, c, d, e\} = \{a\}$
  
- ▶ I læreboka brukes minustegn for differanse:  
 $\{a, b, c\} - \{c, d, e, f\} = \{a, b\}$

# Differanse

Eksempel på differanse mellom relasjoner:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Lise	79023365	Vei 238-5
Geir	25364758	Viken 2

## Medlem

navn	adr	tlf
Lise	Bakken 1	37485960
Liv	Lia 3	19203142
Jan	Heia 7	26374859
Geir	Viken 2	25364758

## Ansatt \ Medlem

navn	tlf	adr
Ali	22465819	Lia 3
Lise	79023365	Vei 238-5

# 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 tilfredsstiller betingelsen  $C$
- ▶  $C$  er et vilkårlig boolsk uttrykk bygget opp fra atomer på formen **op**<sub>1</sub> $\theta$  **op**<sub>2</sub> der
- ▶ operandene **op**<sub>1</sub> og **op**<sub>2</sub> er
  - ▶ enten to attributter i  $R$  med samme domene
  - ▶ eller ett attributt i  $R$  og en konstant fra dette attributtets domene
- ▶ operatoren  $\theta \in \{=, \neq, <, >, \leq, \geq, \text{like}\}$

# Seleksjon

Eksempel på seleksjon:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Geir	25364758	Viken 2

$\sigma_{\text{tlf} > 24000000}$  (Ansatt)

navn	tlf	adr
Lise	37485960	Bakken 1
Geir	25364758	Viken 2



# 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  $\pi_L(R)$

# Projeksjon

Eksempel på projeksjon:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Geir	25364758	Viken 2

$\pi_{\text{navn,adr}}(\text{Ansatt})$

navn	adr
Ali	Lia 3
Lise	Bakken 1
Geir	Viken 2

# Operatorer som spleiser tupler

- ▶ Kartesisk produkt:  $R \times S$
- ▶ Join:  $R \bowtie_C S$  (der  $c$  er joinbetingelsen)
- ▶ Naturlig join:  $R \bowtie S$

# 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 konkatenerert til et tuppel  $v = t \cdot u$  i  $R \times S$
- ▶ I resultatskjemaet løses eventuell navnelikhet mellom attributter i  $R$  og  $S$  ved å kvalifisere navnene med opprinnelsesrelasjonen:  $R.A$ ,  $S.B$
- ▶ Hvis  $R$  og  $S$  er samme relasjon, må en av dem først renavnes

## Kartesisk produkt

### Spisested

snavn	mkat
A	kosher
A	vegetabilsk
B	uten melk
B	hallal
B	glutenfri
B	kosher
C	glutenfri
C	hallal
C	kosher
D	vegetabilsk

### Menykrav

navn	mkat
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

### Spisested × Menykrav

snavn	Spisested.mkat	navn	Menykrav.mkat
A	kosher	Ali	hallal
A	vegetabilsk	Ali	hallal
B	uten melk	Ali	hallal
B	hallal	Ali	hallal
B	glutenfri	Ali	hallal
B	kosher	Ali	hallal
C	glutenfri	Ali	hallal
C	hallal	Ali	hallal
C	kosher	Ali	hallal
D	vegetabilsk	Ali	hallal
A	kosher	Liv	kosher
A	vegetabilsk	Liv	kosher
B	uten melk	Liv	kosher
B	hallal	Liv	kosher
B	glutenfri	Liv	kosher
B	kosher	Liv	kosher
C	glutenfri	Liv	kosher
C	hallal	Liv	kosher
C	kosher	Liv	kosher
D	vegetabilsk	Liv	kosher
A	kosher	Lise	kosher
A	vegetabilsk	Lise	kosher
B	uten melk	Lise	kosher
B	hallal	Lise	kosher
B	glutenfri	Lise	kosher
B	kosher	Lise	kosher
C	glutenfri	Lise	kosher
C	hallal	Lise	kosher
C	kosher	Lise	kosher
D	vegetabilsk	Lise	kosher
A	kosher	Geir	glutenfri
A	vegetabilsk	Geir	glutenfri
B	uten melk	Geir	glutenfri
B	hallal	Geir	glutenfri
B	glutenfri	Geir	glutenfri
B	kosher	Geir	glutenfri
C	glutenfri	Geir	glutenfri
C	hallal	Geir	glutenfri
C	kosher	Geir	glutenfri
D	vegetabilsk	Geir	glutenfri

# Join

- ▶ Intuitivt: Skjøte sammen to relasjoner

$$R \bowtie_C S$$

- ▶ Intuitivt:
  - ▶ 1. Beregn  $R \times S$
  - ▶ 2. Velg ut de tuplene som tilfredsstiller joinbetingelsen  $C$

## Join

### Bistro

bn	mkat
A	kosher
A	vegetabilsk
B	uten melk
B	hallal
B	glutenfri
B	kosher
C	glutenfri
C	hallal
C	kosher
D	vegetabilsk

### Krav

navn	mkat
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

## Bistro $\bowtie$ Krav

C

bn	Bistro.mkat	navn	Krav.mkat
B	hallal	Ali	hallal
C	hallal	Ali	hallal
A	kosher	Liv	kosher
B	kosher	Liv	kosher
C	kosher	Liv	kosher
A	kosher	Lise	kosher
B	kosher	Lise	kosher
C	kosher	Lise	kosher
B	glutenfri	Geir	glutenfri
C	glutenfri	Geir	glutenfri

C: Bistro.mkat = Krav.mkat

## Naturlig join

- ▶  $R \bowtie S$  Ser 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 én gang i de sammensmeltede attributtene
  - ▶ Resultatskjemaet har attributtene i  $R$  etterfulgt av de attributtene i  $S$  som ikke også forekommer i  $R$



# Naturlig join

## Bistro

bn	mkat
A	kosher
A	vegetabilsk
B	uten melk
B	hallal
B	glutenfri
B	kosher
C	glutenfri
C	hallal
C	kosher
D	vegetabilsk

## Krav

navn	mkat
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

## Bistro $\bowtie$ Krav

bn	mkat	navn
B	hallal	Ali
C	hallal	Ali
A	kosher	Liv
B	kosher	Liv
C	kosher	Liv
A	kosher	Lise
B	kosher	Lise
C	kosher	Lise
B	glutenfri	Geir
C	glutenfri	Geir

# Hengetupler

- ▶ Et **hengetuppel** 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

# Hengetupler

## Bistro

bn	mkat
A	kosher
A	vegetabilsk
B	uten melk
B	hallal
B	glutenfri
B	kosher
C	glutenfri
C	hallal
C	kosher
D	vegetabilsk

## Krav

navn	mkat
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

Hengetupler satt med  
grå typer i  
Bistro-relasjonen

## Bistro × Krav

bn	mkat	navn
B	hallal	Ali
C	hallal	Ali
A	kosher	Liv
B	kosher	Liv
C	kosher	Liv
A	kosher	Lise
B	kosher	Lise
C	kosher	Lise
B	glutenfri	Geir
C	glutenfri	Geir

# 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

# Renavning

## Menykrav

navn	matkategori
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

$\rho_{\text{Krav}}(\text{navn}, \text{begrensning})$  (Menykrav)

har som resultat:

## Krav

navn	begrensning
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

# Divisjon

- ▶ Gitt to relasjoner  
 $R(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$  og  
 $S(B_1, B_2, \dots, B_m)$
- ▶  $R \text{ div } S$  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 bruk av divisjon

**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	hallal
B	glutenfri
B	kosher
C	glutenfri
C	hallal
C	kosher
D	vegetabilsk

### Menykrav

navn	krav
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

# Spisested som dekker alles krav til menyen

$\rho_{\text{Allekrav}(\text{matkategori})}(\pi_{\text{krav}}(\text{Menykrav}))$

$\rho_{\text{MuligeSpisesteder}}(\text{Spisested } \mathbf{div} \text{ AlleKrav})$

Kan også uttrykkes som

$\text{Allekrav}(\text{matkategori}) \leftarrow \pi_{\text{krav}}(\text{Menykrav})$

$\text{MuligeSpisesteder}(\text{navn}) \leftarrow \text{Spisested } \mathbf{div} \text{ AlleKrav}$

der  $S(A_1, A_2, \dots, A_n) \leftarrow R(B_1, B_2, \dots, B_n)$  resulterer i relasjonen  $S$  med attributnavnen  $A_1, A_2, \dots, A_n$ .



# Spisested som dekker alle krav til menyen

$$\rho_{\text{Allekrav}}(\text{matkategori}) (\pi_{\text{krav}}(\text{Menykrav}))$$
$$\rho_{\text{MuligeSpisesteder}}(\text{Spisested} \text{ div } \text{Allekrav})$$

## Spisested

navn	matkategori
A	kosher
A	vegetabilsk
B	uten melk
B	hallal
B	glutenfri
B	kosher
C	glutenfri
C	hallal
C	kosher
D	vegetabilsk

## Menykrav

navn	krav
Ali	hallal
Liv	kosher
Lise	kosher
Geir	glutenfri

## Allekrav

matkategori
hallal
kosher
glutenfri

## MuligeSpisesteder

navn
B
C

# Snitt, join og divisjon uttrykt ved de andre operatorene

- ▶  $R \cap S = R \setminus (R \setminus S)$
- ▶  $R \underset{\text{joinbetingelse}}{\bowtie} S = \sigma_{\text{joinbetingelse}}(R \times S)$
- ▶  $R \bowtie S = \pi_{R \cup (S \setminus R)}(\sigma_{R.R \cap S = S.R \cap S}(R \times S))$
- ▶  $R(A, B) \text{ div } S(B) = \pi_A(R) \setminus \pi_A((\pi_A(R) \times S) \setminus R)$

# Bag

- ▶ Kommersielle DBMSer benytter *bag* og ikke *set* (mengde) som grunntype for å realisere relasjoner
- ▶ SQL beregner bager (med unntak av noen av operatorene)
  - ▶ 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  
 $\{a, b, c\} = \{a, c, b\} \neq \{a, a, b, c\} = \{c, a, b, a\} \neq \{a, b, b, c\}$

## Hvorfor *bag* og ikke *set*

- ▶ *Bag* gir mer effektive beregninger av union og projeksjon enn *set*
- ▶ Ved aggregering (foreleses i neste forelesning) 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*
  - ▶ Eksempel:  $(R \cup S) \setminus T = (R \setminus T) \cup (S \setminus T)$   
er riktig for *set*, men ikke alltid for *bag*
- ▶ Når vi på de følgende lysarkene 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  $R$
- ▶ Eksempel på vanlig bagunion:  
 $\{a, a, b, c, c\} \cup \{a, c, c, c, d\} = \{a, a, a, b, c, c, c, c, d\}$

# Bagunion

- ▶ Eksempel på *bagunion* av relasjoner:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Geir	25364758	Viken 2

## Medlem

navn	adr	tlf
Lise	Bakken 1	37485960
Liv	Lia 3	19203142
Jan	Heia 7	26374859
Geir	Viken 2	25364758

## Ansatt $\cup$ Medlem

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Lise	37485960	Bakken 1
Liv	19203142	Lia 3
Jan	26374859	Heia 7
Geir	25364758	Viken 2
Geir	25364758	Viken 2

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



# Bagsnitt

Eksempel på *bagsnitt* mellom relasjoner:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Lise	37485960	Bakken 1
Geir	25364758	Viken 2
Geir	25364758	Viken 2

## Medlem

navn	adr	tlf
Lise	Bakken 1	37485960
Lise	Bakken 1	37485960
Liv	Lia 3	19203142
Jan	Heia 7	26374859
Geir	Viken 2	25364758

## Ansatt $\cap$ Medlem

navn	tlf	adr
Lise	37485960	Bakken 1
Lise	37485960	Bakken 1
Geir	25364758	Viken 2

# 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  $R \setminus S$
- ▶ Eksempel på vanlig bagdifferanse:  
 $\{a, a, b, c, c\} \setminus \{a, c, c, c, d\} = \{a, b\}$

# Bagdifferanse

Eksempel på *bagdifferanse* mellom relasjoner:

## Ansatt

navn	tlf	adr
Ali	22465819	Lia 3
Lise	37485960	Bakken 1
Geir	25364758	Viken 2
Geir	25364758	Viken 2

## Medlem

navn	adr	tlf
Lise	Bakken 1	37485960
Liv	Lia 3	19203142
Jan	Heia 7	26374859
Geir	Viken 2	25364758

## Ansatt \ Medlem

navn	tlf	adr
Ali	22465819	Lia 3
Geir	Viken 2	25364758

## Bagseleksjon

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

## Bagprojeksjon

- ▶ Hvis  $R$  er en bagrelasjon og  $L$  er en (ikketom) 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  $R$

## Kartesisk 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  $n \cdot m$  tupler i  $R \times S$

## Join på bager

- ▶ Hvis  $R$  og  $S$  er bagrelasjoner, fremkommer bagrelasjonen  $R \bowtie_{\text{joinbetingelse}} S$  slik:
  - ▶ Beregn  $R \times S$  (kartesisk produkt av bager)
  - ▶ Velg ut de tuplene som tilfredsstiller betingelsen
- ▶ Merk: Dette er en matematisk definisjon. I praksis beregnes join-en uten å beregne  $R \times S$

## Naturlig join på bager

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



# Relasjonsalgebratolkning av select-setningen—1

1. Ta det kartesiske produktet av relasjonene i **from**
2. Seleker ifølge betingelsene i **where**
3. Projiser ifølge betingelsene i **select**
4. Fjern flerforekomster hvis **select distinct**
5. Sorter i henhold til **order by**

# Relasjonsalgebra tolkning av select-setningen—1. Eksempel

- ▶ Skjema:  
Spisested(snavn, matkategori)  
Menykrav(navn, matkategori)
- ▶ Query:  

```
select distinct snavn  
from Spisested, Menykrav  
where (navn='Ali' or navn='Liv')  
      and Spisested.matkategori =  
      Menykrav.matkategori  
order by snavn;
```
- ▶  $\tau_{\text{snavn}}(\delta(\pi_{\text{snavn}}(\sigma_c(\text{Spisested} \times \text{Menykrav}))))$
- ▶ Betingelsen  $c$ : (navn = 'Ali' or navn = 'Liv') and Spisested.matkategori = Menykrav.matkategori
- ▶  $\delta(R)$  fjerner flerforekomster av tupler fra bagrelasjonen  $R$
- ▶  $\tau_{A_1, A_2, \dots, A_n}(R)$  sorterer tuplene i  $R$  etter verdiene i  $A_1$ , for like verdier i  $A_1$  sorteres tuplene etter verdiene i  $A_2$ , osv.

# Relasjonsalgebrafortolkning av select-setningen—2

- ▶ Hvis vi ser nøyere på vanlige select-setninger, er det ofte to vesensforskjellige typer setningsledd i where-klausulen:
  - ▶ *Seleksjon* av tupler i en relasjon
  - ▶ *Join* av tupler på tvers av relasjoner
- ▶ Selv om den forenklete tolkningen av select-setningen er riktig matematisk sett, vil databasesystemet velge en tolkning som er mer effektivt beregnbar

## Relasjonsalgebratolkning av select-setningen—2 (forts.)

Derfor vil databasesystemet oversette til et uttrykk som likner det vi får ved å gjøre følgende:

1. Seleker ifølge *seleksjonsbetingelsene* i **where** (gir færre tupler i relasjonene før join)
2. Join relasjonene i **from** i henhold til *joinbetingelsene* i **where**
3. Projiser ifølge betingelsene i **select**
4. Fjern flerforekomster hvis **select distinct**
5. Sorter i henhold til **order by**

# Relasjonsalgebratolkning av select-setningen—2. eksempel

- ▶ Skjema:  
Spisested(snavn, matkategori)  
Menykrav(navn, matkategori)

- ▶ Query:

```
select distinct snavn  
from Spisested, Menykrav  
where (navn='Ali' or navn='Liv')  
      and Spisested.matkategori =  
        Menykrav.matkategori  
order by snavn;
```

- ▶  $\tau_{\text{snavn}}(\delta(\pi_{\text{snavn}}(\text{Spisested} \bowtie_{\text{jb}} \sigma_{\text{sb}}(\text{Menykrav}))))$
- ▶ Seleksjonsbetingelsen sb: navn = 'Ali' or navn = 'Liv'  
Joinbetingelsen jb: Spisested.matkategori = Menykrav.matkategori
- ▶ I dette tilfellet kunne joinbetingelsen alternativt vært oversatt til naturlig join

## Klassisk union, snitt, differanse og kartesisk produkt uttrykt i SQL

<i>Algebra</i>	<i>SQL</i>
$R \cup S$	<b>select * from R union S;</b>
$R \cap S$	<b>select * from R intersect S;</b>
$R \setminus S$	<b>select * from R except S;</b>
$R \times S$	<b>select * from R cross join S;</b>
$R \times S$	<b>select * from R, S;</b>

*Husk krav om unionkompatibilitet for R og S i union, intersect, except!*

## Bagunion, bagsnitt og bagdifferanse uttrykt i SQL

<i>Algebra</i>	<i>SQL</i>
$R \cup S$	<b>select * from R union all S;</b>
$R \cap S$	<b>select * from R intersect all S;</b>
$R \setminus S$	<b>select * from R except all S;</b>

*Husk krav om unionkompatibilitet for R og S i* **union all, intersect all, except all!**

# Kostbare operasjoner i SQL

- ▶ **distinct:**
  - ▶ Sortering er generelt kostbart
  - ▶ Bør brukes med forsiktighet
- ▶ **union, intersect, except:**
  - ▶ SQL beregner set-variantene av disse (dvs. at flerforekomster fjernes)
  - ▶ Vurder å bruke **union all**, **intersect all**, **except all** som er bag-variantene