

Dokumentet viser når det er hensiktsmessig å benytte av de ulike symbolene (Selection(σ), Projection(π), Kartesisk Produkt($R \times R$), Union(\cup), Difference($R_1 - R_2$), Join(\bowtie) og Renaming(ρ).

Selection(σ)

Dette tegnet henter ut de tuplene som oppfyller en gitt betingelse.

Selection

- Return all tuples that satisfy a condition
- Notation: $\sigma_C(R)$
 - C: boolean condition that output tuples should satisfy
 - =, \neq , <, >, \geq , \leq , AND, OR, NOT, ...
 - R: input relation
 - Output schema: same as input schema (i.e., R's schema)
- Selection σ is commutative
$$\sigma_{C_1}(\sigma_{C_2}(R)) = \sigma_{C_2}(\sigma_{C_1}(R))$$
 - A sequence of select operations may be applied in any order
 - A cascade of select operations may be replaced by a single selection with a conjunction of all the conditions

Selection example

$\sigma_{Age > 22}(\text{Employee})$

EID	Name	Age	Department
6	Ann	24	SD
25	John	23	IT

$\sigma_{Age > 22}$

Employee			
EID	Name	Age	Department
19	Ann	24	SD
6	Ann	24	SD
3	Paul	23	CS
25	John	23	IT
5	Peter	21	SD

I bildet over, så ser vi at vi velger ut de tuplene som tilfredsstiller betingelsen hvor vi plukker ut EMPLOYEE-tupler som har en alder over 22.

Projection(π)

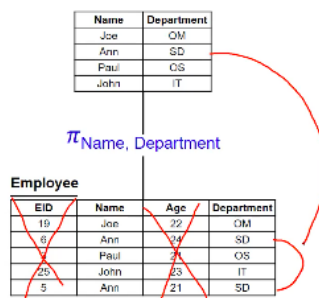
Dette tegnet plukker ut de tuplene som vi er ute etter å hente.

Projection

- Return specific attributes of all tuples
- Notation: $\pi_{A_1, \dots, A_n}(R)$
 - Input schema: $R(B_1, \dots, B_m)$
 - A_1, \dots, A_n : list of attributes to project onto, s.t. $\{A_1, \dots, A_n\} \subseteq \{B_1, \dots, B_m\}$
 - Output schema: $S(A_1, \dots, A_n)$
- The project operation *removes any duplicate tuples* (the result of project is a relation, i.e., a set of tuples)
- Projection is *not* commutative

Projection example

$\pi_{\text{Name, Department}}(\text{Employee})$



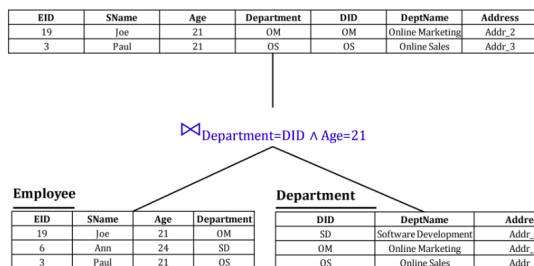
F.eks så henter vi ut tuplene som kun har NAME og DEPARTEMENT fra EMPLOYEE-tabellen.

Join(\bowtie)

Fungerer akkurat som ved SQL hvor vi “JOINER” sammen to ulike tabeller i ett.

Theta Join example

$\text{Employee} \bowtie_{\text{Department}=\text{DID} \wedge \text{Age}=21} \text{Department}$



Equi-join example

Employee $\bowtie_{\text{Department}=DID}$ Department

EID	Name	Age	Department	DID	DeptName	Address
19	Joe	21	OM	OM	Online Marketing	Addr_2
6	Ann	24	SD	SD	Software Development	Addr_1
3	Paul	21	OS	OS	Online Sales	Addr_3

$\bowtie_{\text{Department}=DID}$

Employee

EID	Name	Age	Department
19	Joe	21	OM
6	Ann	24	SD
3	Paul	21	OS

Department

DID	DeptName	Address
SD	Software Development	Addr_1
OM	Online Marketing	Addr_2
OS	Online Sales	Addr_3

Natural join example

Employee \bowtie Department

$= \pi_{\text{EID, Name, Age, DID, DeptName, Address}} (\sigma_{DID=DID2} (\text{Employee} \times \rho_{DID2, DeptName, Address} (\text{Department})))$

EID	Name	Age	DID	DeptName	Address
19	Joe	21	OM	Online Marketing	Addr_2
6	Ann	24	SD	Software Development	Addr_1
3	Paul	21	OS	Online Sales	Addr_3

\bowtie

Employee

EID	Name	Age	DID
19	Joe	21	OM
6	Ann	24	SD
3	Paul	21	OS

Department

DID	DeptName	Address
SD	Software Development	Addr_1
OM	Online Marketing	Addr_2
OS	Online Sales	Addr_3

Renaming(ρ)

Det er å skrive om kolonnene i en gitt tabell. Hvis vi ser over på EMPLOYEE, så har de en NAVN-kolonne. Vi kan skrive om denne kolonnen fra å hete NAVN -> UNAVN. Hensikten med dette skyldes av at ved relasjonsalgebra så må vi skille mellom diverse kolonner på tvers av tabeller. Vi kan ha to ulike tabeller med like kolonnenavn, men måten vi skiller dem gjennom relasjonsalgebra, er ved å benytte av Renaming(ρ).

Selection(σ)

Projection(π)

Kartesisk Produkt($R \times R$)

Union(\cup)

Difference($R1 - R2$)

Join(\bowtie)

Renaming(ρ)

Student

Navn	Fødselsdato	StudentNr	Adresse	Studie
Ola	1997-01-01	1234	Sognsveien 1	Informatikk
Kari	1995-10-12	1032	Sognsveien 1	Matematikk
Sara	1991-03-09	2501	Osloveien 4	Informatikk
Per		4510	Sognsveien 1	Informatikk

Gruppelærer

StudentNr	Kurs	Semester
1234	IN2090	H18
2501	IN2090	H18
2501	IN1234	V19

Navn på de studentene som bor i sognsveien 1

π Navn(σ Adresse = Sognsveien 1(Student))

(Plukker ut navnene gjennom π , Henter ut de adressene som tilsvarer Sognsveien 1 gjennom σ)

Studentnr på alle gruppelærerne i IN2090 høsten 2018

π StudentNr(σ (Kurs = IN2090 \wedge Semester = H18(Gruppelærer)))

(Plukker ut studentNr gjennom π , Henter ut Gruppelærere som studerte i IN2090 for høsten 2018 gjennom σ . Bruker \wedge tegnet som understreker "OG")

StudentNr til alle studenter som studerer informatikk

π StudentNr(σ Studie = Informatikk(Student))

For alle studenter som bor i Sognsveien 1, finn de som studerer Matematikk

π Navn(σ Adresse = Sognsveien 1(Student) \wedge σ Studie = 'Matematikk')

Finn adressen til alle studenter som har et studentNR mellom 1000 og 3000

π Adresse(σ StudentNr > 1000 \wedge StudentNr < 3000)

Navn og fødselsdato på alle gruppelærerne i IN2090 høsten 2018

π Navn, Fødselsdato(σ (Kurs = IN2090 \wedge Semester = H18(Gruppelærer) \bowtie Student))

Skriv et uttrykk i relasjonsalgebraen som finner navn på alle barn som ønsker seg hoppestokk.

π Navn(barn bid \bowtie ønskeliste \bowtie Gave gid π gid(σ navn = 'hoppestokk')(Gave))