ALGEBRA RELAȚIONALĂ

Proiecția (π)

- $L = (a_1,\ldots,a_n)$ – listă de atribute ale relației R
- Proiecția returnează o relație eliminând toate atributele care nu sunt în L

$$\pi_L(R) = \{ t \mid t_1 \in R \wedge t.a_1 = t_1.a_1 \wedge \ldots \wedge t.a_n = t_1.a_n\}$$

EX:

$\pi_{cid,\ grade}$(Enrolled)

$\pi_{cid,\ grade}$(

sid	cid	grade
1234	Alg1	9
1235	Alg1	10
1234	DB1	10
1234	DB2	9
1236	DB1	7
1237	DB2	9
1237	DB1	5
1237	Alg1	10

) =

cid	grade
Alg1	9
Alg1	10
DB1	10
DB2	9
DB1	7
DB1	5

- Algebra relațională operează cu mulțimi

⇨ $\pi_{cid,\ grade}$(Enrolled) ⇔ SELECT DISTINCT cid, grade
FROM Enrolled

Selecția (σ)

- Selectează tuplurile unei relații R care verifică o condiție c (predicat de selecție)

$$\sigma_c = \{ t \mid t \in R \wedge c\}$$

$\sigma_{grade > 8}$(

sid	cid	grade
1234	Alg1	9
1235	Alg1	10
1234	DB2	9
1236	DB1	7
1237	DB1	5
1237	Alg1	6

) =

sid	cid	grade
1234	Alg1	9
1235	Alg1	10
1234	DB2	9

$\sigma_{grade>8}(Enrolled)$ ⇔ SELECT DISTINCT *
FROM Enrolled
WHERE grade > 8

Compunerea

$\pi_{cid,\ grade}(\sigma_{\ grade\ >\ 8}($

sid	*cid*	*grade*
1234	Alg1	9
1235	Alg1	10
1234	DB1	10
1234	DB2	9
1236	DB1	7
1237	DB2	9
1237	DB1	5
1237	Alg1	10

$)) =$

cid	*grade*
Alg1	9
Alg1	10
DB1	10
DB2	9

Reuniunea

$R_1 \cup R_2$ ⇔ SELECT DISTINCT * FROM R_1
UNION
SELECT DISTINCT * FROM R_2

Intersecția

$R_1 \cap R_2$ ⇔ SELECT DISTINCT * FROM R_1
INTERSECT
SELECT DISTINCT * FROM R_2

Diferența

$R_1 \setminus R_2$ ⇔ SELECT DISTINCT * FROM R_1
EXCEPT
SELECT DISTINCT * FROM R_2

Produs cartezian

- Combinarea a două relații $R_1(a_1, \ldots, a_n)$ și $R_2(b_1, \ldots, b_m)$

$$R_1 \times R_2 = \{ t \mid t_1 \in R_1 \wedge t_2 \in R_2 \wedge t.a_1 = t_1.a_1 \wedge \ldots \wedge t.a_n = t_1.a_n \wedge t.b_1 = t_2.b_1 \wedge \ldots \wedge t.b_m = t_2.b_m \}$$

SELECT DISTINCT *

FROM R_1, R_2

θ-Join

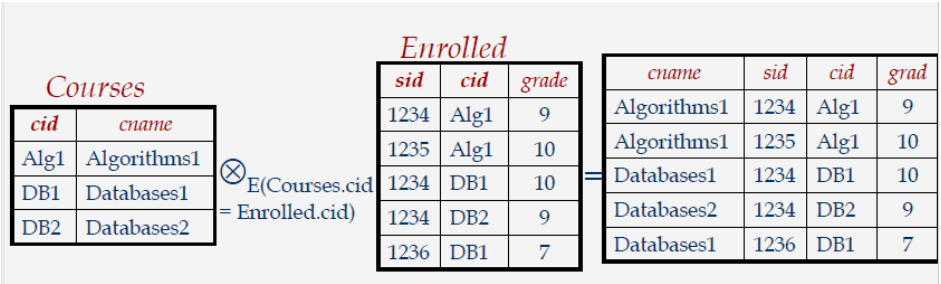
- Combinarea a două relații R_1 și R_2 cu respectarea condiției c

$$R_1 \ (x) \ R_2 = \sigma_c(R_1 \ X \ R_2)$$

- Echivalentă în SQL cu INNER JOIN

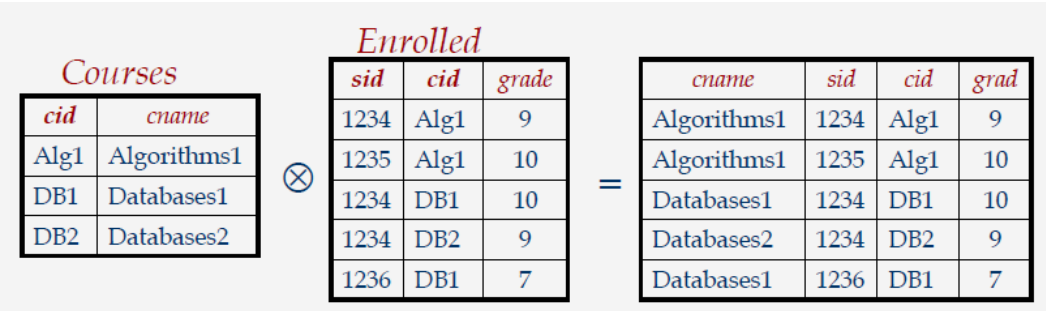
Equi-Join

- Combină două relații pe baza unei condiții compuse doar din egalități ale unor atribute aflate în prima și a doua relație și proiectează doar unul dintre atributele redundante (deoarece sunt egale)â



Join Natural

- Combină două relații pe baza egalității atributelor ce au *același nume* și proiectează doar unul dintre atributele redundante



Câtul

- Nu este un operator de bază, însă simplifică mult interogarea în anumite situații
- Fie R_1 cu 2 atribute, x și y, și R_2 cu un atribut y:

  $R_1 / R_2 = \{<x> \mid \exists <x, y> \in R_1 \; \forall <y> \in R_2\}$

  adică, $R_1 \setminus R_2$ conține toate tuplurile x astfel încât pentru fiecare dintre tuplurile y din R_2, există câte un tuplu xy în R_1

- Generalizând, x și y pot reprezenta orice mulțime de atribute; y este mulțimea atributelor din R_2, și $x \cup y$ reprezintă atributele lui R_1.
- Câtul nu este operator esențial, ci doar o "scurtătură"
- *Ideea*: Pentru R_1 / R_2, vom determina valorile x care nu sunt 'conectate' cu anumite valori y din R_2 (valoarea x este deconectată dacă, atașând la ea o valoare y din R_2, obținem un tuplu xy ce nu se regăsește în R_1)

Valorile x deconectate: $R_1 / R_2 = \pi_x(R_1)$

Redenumirea

- Dacă atributele și relațiile au același nume este necesar să putem redenumi una din ele

$$\rho(R' (N_1 \rightarrow N'_1, N_2 \rightarrow N'_2), R)$$

$$\rho(\text{Courses2}\ (\text{cid} \rightarrow \text{code},\ \text{cname} \rightarrow \text{description}),\ \text{Courses})$$

Courses

cid	*cname*	*credits*
Alg1	Algorithms1	7
DB1	Databases1	6
DB2	Databases2	6

→

Courses2

code	*description*	*credits*
Alg1	Algorithms1	7
DB1	Databases1	6
DB2	Databases2	6

```
SELECT cid as code,
       cname as description,
       credits
FROM Courses Courses2
```