Ex: List of all possible pairs of math & CS majors.

get a relation w/ all math majors:

$\rho_{\text{Math Maj}}(\Pi_{\text{name}}(\sigma_{\text{major}=\text{math}}(\text{Students})))$ ①

Name
Bob
Eva

get CS majors

$\rho_{\text{CS majors}}(\Pi_{\text{name}}(\sigma_{\text{major}=\text{CS}}(\text{Students})))$ ②

Name
Alice
Carol
Dan

→ ① × ②

Mathmajor	CSmajor
Bob	Alice
Bob	Carol
Bob	Dan
Eve	A
Eve	C
Eve	D

Natural join operator ⋈

↳ Similar to a cross product.

Diff: Only matches rows in the 2 tables where all common attributes are identical.

Cartesian product ×

Nat join ⋈

$R \bowtie S$ ① Find the attributes in common btw R & S.

② Make a new relation w/ all the attribs from R & S. (no duplicates)

③ Add rows to the new relation for every pair of rows from R, S but only where the common attribs. match up.

Example

I want a list of all the CRNs Alice is enrolled in.

$\rightarrow \Pi_{CRN}\left(\sigma_{Name=Alice}\left(Enrolled \bowtie Students\right)\right)$ ~~$\sigma_{ID=1}(Enrolled)$~~

ID	CRN	Name	major	Use

$\rightarrow \Pi_{CRN}\left(\left(\sigma_{Name=Alice}(Students)\right) \bowtie Enrolled\right)$

A list of course names Alice is enrolled in.

$\Pi_{course\ name}\left(\left[\text{Take either query above}\right] \bowtie Courses\right)$

$\Pi_{course\ name}\left(\sigma_{name=Alice}\left(Stu \bowtie Enrolled \bowtie Courses\right)\right)$

Theta join $\bowtie_{\theta}$: condition on which to join the tables

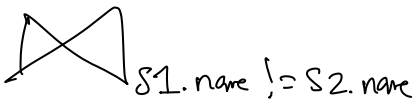
Ex: A list of students, with their majors, and all the courses taught in the same dept as their major.

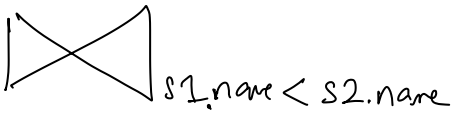
$\Pi_{name,\ dept,\ coursename}\left(Students \bowtie_{major=dept} Courses\right)$

Q: List of all possible pairs of CS majors.

$$\rho_{S1}\left(\sigma_{major=CS}(Students)\right) \times \rho_{S2}\left(\sigma_{major=CS}(Students)\right)$$

theta join





~~Bob~~

Alice	Carl
Carl	Alice
~~Alice~~	~~Alice~~

Alice	Carl
Alice	Dan
Carl	Dan

Theta join can be written as Cartesian product.

$$R \bowtie_\theta S \longleftrightarrow \sigma_\theta(R \times S)$$

Shortcut notation / Linear notation

Allows you to give temporary names to relations & use those names in later queries.

$$CS := \sigma_{major=CS}(Students)$$

$$\rho_{CS1}(CS) \times \rho_{CS2}(CS)$$

(1) All pizzerias frequented by at least one person who is under 18.

(2) Find names of all Rhodes students who eat either mushroom or pepperoni pizza (or both)

(3) " " " " " " " " eat mushroom pizza AND pepperoni pizza.

---

(1) $\Pi_{pizzeria}\left(\sigma_{age<18}(Person \bowtie Frequents)\right)$

$\Pi_{pizzeria}\left(\sigma_{age<18}(Person) \bowtie Frequents\right)$

(2) $\Pi_{name}\left(\sigma_{school = Rhodes \wedge (pizza = mushroom \vee pizza = pepperoni)}(Person \bowtie Eats)\right)$

changing this to "∧" doesn't work

(3) if (pizza == "mush" && pizza == "pepp") ← this doesn't make sense

$\Pi_{name}\left(\sigma_{school=Rhodes \wedge pizza=mush}(P \bowtie E)\right) \cap \Pi_{name}\left(\sigma_{school=Rhodes \wedge pizza=pepp}(P \bowtie E)\right)$

Eats(name, likesmush, likespepp, likesveg)

Majors(~~S~~R#, major1, major2, ...) ← AWFUL