Relational Algebra

Language for querying DBs.

Why?
- Easier to write, to learn
- Easy to optimize (query optimizer)

Similar to boolean algebra
operates on T/F values

$p \wedge q$
$(s \vee r) \to a$

R.A. operates on relations

"Regular" algebra
Boolean algebra } operators
↳ $\wedge$, $\vee$, $\neg$, $\sim$, $\to$

R.A. operators

Assume we have 2 relations, called R & S.
" that R & S have the same schema. (same columns)

3 operators from set theory
- Union: $R \cup S$: new relation w/ all rows of R followed by all rows of S
- Intersection: $R \cap S$: new relation w/ all rows that R & S have in common
- Difference: $R - S$: new relation w/ all rows of R that don't appear in S.

Ex: Students $\cup$ Professors

#rows?
8

Last	First

Students $\cap$ Profs

Last	First
Longbottom	Neville

Gryffindors − Students #rows? 2

Students − Gryffindors #rows? 1

Projection operator Π

↳ Takes a relation ⟶ gives a new relation w/ only certain columns.

$\Pi_{A_1, A_2, A_3 \ldots}(R)$ ⟶ make a new relation from R with only attributes $A_1, A_2, A_3 \ldots$

Ex: $\Pi_{Name}(\text{Students})$:

Name
Alice
Bob
C
D
E
F

$\Pi_{major}(\text{Students})$:

major
CS
math
physics

$\Pi_{dept}(\text{courses})$: # rows? 4

$\Pi_{dept, seats}(\text{Courses})$:

depts	seats
CS	20
CS	15
⋮	⋮

9 rows

σ Signma operator / Selection operator : takes a relation & gives a new rel'n w/ only certain ~~c~~ rows

$\sigma_{condition}(R)$: give me all rows of R where the condition is true.

$\sigma_{major = cs}(\text{Students})$ ⟶

Name	ID	major	Age
Alice			
Carol			
Dan			

$\sigma_{dept = music}(\text{Courses})$

Rewrite without using "$\wedge$" ⟶ $\sigma_{major = CS \wedge age \leq 19}(\text{Students})$ Alice & Carol

$\sigma_{major=cs}(\text{Students}) \cap \sigma_{age \le 19}(\text{Students})$

$\sigma_{major=cs}(\sigma_{age \le 19}(\text{Students}))$

```
if (x && y)
{
  ---
}

if (x)
{  if (y)
   {
     ---
   }
}
```

Rho ρ – renaming (relation or attributes)

$\rho_{S(A_1, A_2, A_3 \ldots)}(R)$ – take relation R, make a new relation called S w/ same columns as R, but renamed to $A_1, A_2, A_3, \ldots$

$\rho_S(R)$ – take R & make a copy of it, renamed to S

~~Combining~~ /Chaining

Combining – give me a table of all the names of CS majors

$\Pi_{name}(\sigma_{major=cs}(\text{Students}))$

$\sigma_{major=cs}(\underbrace{\Pi_{name}(\text{Students})}_{\text{Doesn't work}})$ — $\boxed{\text{Name}}$

– give me CRN & names of all math or CS courses w/ < 20 seats.

$\Pi_{crn, name}(\sigma_{(dept=CS \lor dept=math) \land seats<20}(\text{Courses}))$

Combining Multiple Relations

– Cartesian product

$R \times S$: Make a new relation pairing each row of R w/ every possible row in S.

New rel'n always has (# rows in R) × (# rows in S)

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

get a relation w/ all math majors:

$$\underbrace{\Pi_{name}\left(\sigma_{major=math}(Students)\right)}_{1}$$

Name
Bob
Eva

get CS majors

$$\underbrace{\Pi_{name}\left(\sigma_{major=CS}(Students)\right)}_{2}$$

Name
Alice
Carol
Dan

$\longrightarrow$ 1 × 2