Students (RIN, Email, Name, Address, GPA, DOB)

Key: RIN
Key: Email

Key: NAME, Address, DOB

(RIN, ~~Name~~) ← unique not minimal

Key is a minimal set of attributes such that no two tuples can have the same value for the key

---

R(A, B, C, D)

Key: A
Key: B, C
Key: B, D

RELATIONAL ALGEBRA

1 SELECTION

$\sigma_C(R)$ SELECT$_C$(R)

input: R, C over attrs of R → output: A new relation w/ same schema as R, But only tuples in R that satisfy C

2 PROJECTION

$\pi_{A_1,\dots,A_n}(R)$ PROJECT$_{A_1,\dots,A_n}$(R)

Input: R, $A_1,\dots,A_n$ attrs of R → Output: A tuples in R only attrs $A_1,\dots,A_n$

$$\sigma_{\text{Doors}=4 \text{ AND isAWD}=\text{True}}(\text{carTypes})$$

$$\pi_{\text{make}}\left(\sigma_{\text{Doors}=4 \text{ AND isAWD}=\text{True}}(\text{carTypes})\right)$$

Make
Kia
Rivian
Tesla

Find all cars owned by students registered in NY, return their license

$$R1 = \sigma_{State=NY} (StudentCars)$$

$$R2 = \pi_{License} (R1)$$

ASD423
RFW424
DYH456
JHY452

(3) RENAME input → A relation Output: same relation but attributes are renamed

$$R1(RIN, L1, S1) = StudentCars(RIN, License, State)$$

$$R1 = \rho_{RIN, L1, S1} (StudentCars)$$

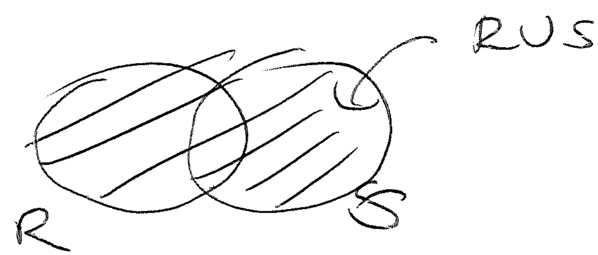
SET OPERATIONS

set compatibility = Two relations (R, S) are set compatible if they have exactly the same schema (same attrs / same names)

SET UNION

$R \cup S$ R UNION S

$R \cup S$ is all tuples in ~~both~~ either relation (tuples in R **OR** tuples in S)

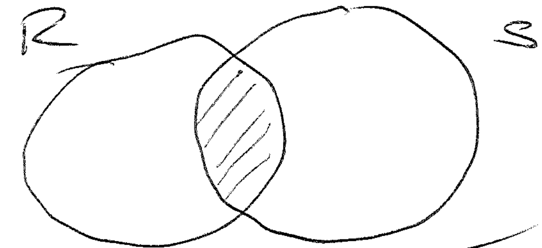


License, state of all cars owned by faculty **OR** students

$$R = \pi_{License,\ state}(StudentCars \cup FacultyCars)$$

SET INTERSECTION $R \cap S$ R INTERSECT S

Tuples in both R <u>and</u> S



Licence, state of all cars owned by ~~a person who is both~~ a faculty and a student

$$\Pi_{License, State} (StudentCars \cap FacultyCars)$$

→ | DHY456, NY |

License, state of all cars owned by a faculty and a student

– License, state of cars owned by a faculty

$$R1 = \Pi_{License, State} (FacultyCars)$$

– License, state of cars owned by a student

$$R2 = \Pi_{License, State} (StudentCars)$$ → | DHY456, NY |
| JHY452, NY |

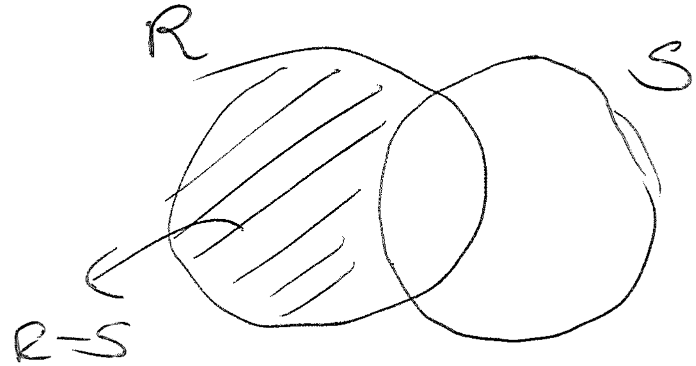
→ Intersect

$$R1 \cap R2$$

SET DIFFERENCE

$R - S$

All tuples in R that are **NOT** in S



All License, State of cars owned by only students (by student but **NOT** faculty)

(cars owned by students) − (cars owned by faculty)

$$\pi_{License, State}(Student\ Cars) - \pi_{License, State}(Faculty\ Cars)$$

6

~~Cars only ow~~

cars in my db no one owns, return license, state

All cars

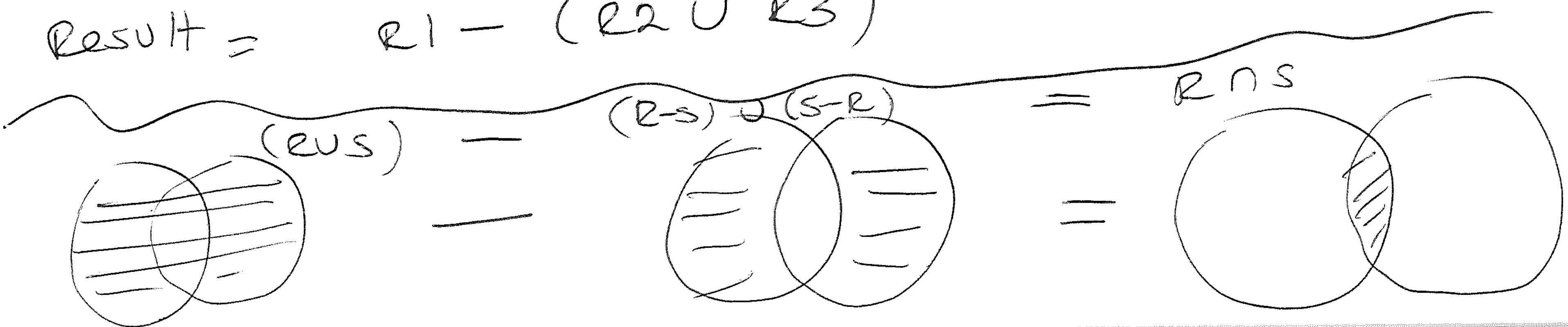
$R1 = \Pi_{License, state} (Cars)$

$R2 = \Pi_{License, state} (StudentCars)$

$R3 = \Pi_{License, state} (FacultyCars)$

$Result = (R1 - R2) - R3$

$Result = R1 - (R2 \cup R3)$



CARTESIAN PRODUCT

R = {(a1, b1), (a1, b2)}

S = { (b1, c2, d1), (b3, c1, d2), (b5, c1, d1)}

R X S = { (a1, b1, b1, c2, d1), (a1, b1, b3, c1, d2), (a1, b1, b5, c1, d1), (a1, b2, b1, c2, d1), (a1, b2, b3, c1, d2), (a1, b2, b5, c1, d1)}

CARTESIAN PRODUCT R X S = T

Output = T has as schema all attributes in R and all attributes in S

and there is a tuple t for every combination of tuples r & s in R & S

t contains all attrs in r, s

Input = R & S have no common attributes

Student RINs for Students who own a red car.

~~R1 = Cars × StudentCars~~

R1 (RIN, L1, S1) = Student cars (54 tuples)

R2 = R1 × Cars

$$R3 = \sigma_{\text{License} = L1 \text{ AND State} = S1 \text{ AND color} = \text{'red'}} (R2)$$

$$R4 = \Pi_{RIN} (R3)$$

$$S1 = \sigma_{\text{color} = \text{'Red'}} (\text{Cars}) \times R1$$

$$S3 = \Pi_{RIN} (S2)$$

$$S2 = \sigma_{\text{License} = L1 \text{ AND State} = S1} (S1)$$

All makes owned by faculty

R1(RIN, L1, S1) = FacultyCars

R2(L2, S2, C2, CI2, M2, V2) = Cars

$$\pi_{Make}\left(\sigma_{L1=L2 \text{ AND } S1=S2 \text{ AND } C2=CarId}\left(CarTypes \times R1 \times R2\right)\right)$$

$$\pi_{MAKE}\left(\left(CarTypes \bowtie_{C2=CarId} R2\right) \bowtie_{L1=L2 \text{ AND } S1=S2} R1\right)$$

THETA JOIN

$R \bowtie_C S$ $\quad$ R $JOIN_C$ S

$$R \bowtie_C S = \sigma_C (R \times S)$$

where C is a join condition comparing attrs for R to S

& R and S have no attrs in common