Key
please check it!
Post questions to Blackboard

STUDENTS IN CLASS GROUP:

Use the handout of three tables -- Vehicle, Own, Person -- to answer these.

0. Assume that Vehicle only has the first two rows shown on the handout and that Person only has the first two rows shown on the handout. Show or otherwise describe the result on the natural join of Vehicle and Person.

Vehicle ⋈ Person							
VRN	Ma	Mo	Color	SSN	Name	Addr	Phone
123	Honda	Hawk	Red	abc	Dave	Birch	xxx
123	Honda	Hawk	Red	bcd	Mary	Grove	yyy
234	Mazda	RX7	Blue	abc	Dave	Birch	xxx
234	Mazda	RX7	Blue	bcd	Mary	Grove	yyy

Important general point: the natural join of two relations/tables that share no same-named attributes is the cross-product of those relations. That is, ALL pairs of rows, one from each table being joined, satisfy the requirement that all values along the ZERO shared attributes are equal.

Give relational algebra expressions for each of the following queries specified in English (assuming the three tables, with all rows, of the handout).

1. Write a query that returns the Name and Phone of all Persons owning VRN=123.

natural join

$$\Pi_{\substack{\text{Name},\\ \text{Phone}}} \left(\sigma_{\substack{\text{VRN}\\ =123}} \left(\text{Own} \bowtie \text{Person} \right) \right)$$

rename operator (rho)

$$\Pi_{\substack{\text{Name},\\ \text{Phone}}} \left(\sigma_{\substack{\text{VRN}\\ =123}} \left(\rho_{\text{Own}(\text{VRN},\text{OSSN})} \text{Own} \bowtie_{\substack{\text{OSSN}\\ =\text{PSSN}}} \rho_{\text{Person}(\text{PSSN},\text{Name}\ldots)} \text{Person} \right) \right)$$

rename SSNs to distinguish them

theta (θ) join makes join condition explicit

This shorthand would be acceptable

$$\Pi_{\substack{\text{Name},\\ \text{Phone}}} \left(\sigma_{\substack{\text{VRN}\\ =123}} \left(\text{Own} \bowtie_{\substack{\text{O.SSN}\\ =\text{P.SSN}}} \text{Person} \right) \right)$$

Alternative "pushes" select inward

$$\Pi_{\substack{\text{Name},\\ \text{Phone}}} \left(\left(\sigma_{\substack{\text{VRN}\\ =123}} \text{Own} \right) \bowtie \text{Person} \right)$$

~~$\Pi_{\substack{\text{Name},\\ \text{Phone}}} \left(\left(\sigma_{\substack{\text{VRN}\\ =123}} \text{Own} \right) \bowtie \left(\Pi_{\text{Name},\ \text{Phone}} \text{Person} \right) \right)$~~ wrong

No basis (SSN) for join

Generally, "pushing" projections almost never good idea

2. Write a query to show the VRN and Mo of each owned Vehicle and the Name and Addr of the Person who owns it.

$\Pi_{VRN, Mo, Name, Addr}((Vehicle \bowtie Own) \bowtie Person)$

$\Pi_{VRN, Mo, Name, Addr}((Vehicle \bowtie (Own \bowtie Person))$

$\Pi_{VRN, Mo, Name, Addr}((Vehicle \bowtie Person) \bowtie Own)$

no (empty) join condition in this case

and of course, can write natural join as theta join that spells out the join condition

$\bowtie_{VRN = VRN \wedge SSN = SSN}$ in this case

or $\Pi_{VRN, Mo, Name, Addr}(Vehicle \bowtie Own \bowtie Person)$

if fine too in this case

3. Write a query to show the VRN and Mo of each owned Vehicle by someone on 'Birch' and the Name of the Person who owns it.

$\Pi_{VRN, Mo, Name}(\sigma_{Addr='Birch'}(Vehicle \bowtie Own \bowtie Person))$

$\Pi_{VRN, Mo, Name}(Vehicle \bowtie Own \bowtie (\sigma_{Addr='Birch'} Person))$ push select inward

$\Pi_{VRN, Mo, Name}(Vehicle \bowtie (Own \bowtie_{SSN=SSN \wedge Addr='Birch'} Person))$

$\wedge$ is 'AND'

I consider this bad style at a minimum, since 'Birch' is not an attribute of one of the tables, but is a constant ~ this might get kicked back from an interpreter of RA ~ try it (but it works on Widem Course RA interpreter)

$Own \bowtie_\theta Person$

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

equivalence of theta join and select/crossproduct speaks to correctness of above

difference operation

for any set operator, two operands must have identical format

4. Write a query that returns the VRN and Mo of any vehicle that isn't owned.

$\Pi_{\substack{VRN,\\ Mo}} Vehicle - \Pi_{\substack{VRN,\\ Mo}} (Vehicle \bowtie Own)$

all vehicles — all owned vehicles

all unowned vehicles

$\Pi_{\substack{VRN,\\ Mo}} \left((\Pi_{VRN} Vehicle - \Pi_{VRN} Own) \bowtie Vehicle \right)$

VRNs of all unowned vehicles

join back to get Mo

I believe this strategy was illustrated in video

rho

5. Write a query that returns pairs of SRNs for (different) Persons that live at the same Addr.

$$\Pi_{\substack{P1.SSN,\\ P2.SSN}}\left(\left(\rho_{P1(SSN\ldots)}\ Person\right) \bowtie_{\substack{P1.Addr\\ =P2.Addr\\ \wedge\ P1.SSN <> P2.SSN}} \left(\rho_{P2(SSN\ldots)}\ Person\right)\right)$$

// SSN comparison here better than name comparison why? In fact, name comparison would be incorrect

OR

$$\begin{cases} P1 := Person \\ P2 := Person \\ \Pi_{\substack{P1.SSN,\\ P2.SSN}}\left(P1 \bowtie_{\substack{P1.Addr=P2.Addr\\ \wedge\ P1.SSN <> P2.SSN}} P2\right) \end{cases}$$

what's wrong with natural join here

$$\Pi_{\substack{P1.SSN,\\ P2.SSN}}\left(\sigma_{\substack{P1.SSN\\ <>\\ P2.SSN}}\left(\left(\rho_{P1(\ldots)}\ Person \bowtie_{\substack{P1.Addr\\ =P2.Addr}} \rho_{P2(\ldots)}\ Person\right)\right)\right)$$

The versions above will all return answers of form

P1.SSN	P2.SSN
123	456
456	123

} conceptual duplication

Best

↳ replace '<>' above, with '<' (or '>')

~~P1.SSN~~
~~P2.SSN~~

only get 123 456

only get 456 123

6. Write a query that returns pairs of Names for (different) Persons that live at the same Addr.

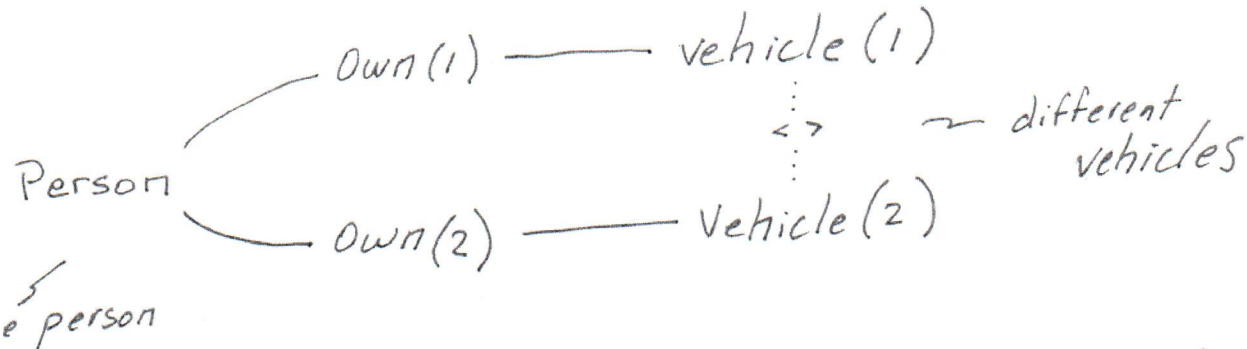
$$\Pi_{P1.Name,\ P2.Name}\left(\left(\left(\rho_{P1(\ldots)}\ Person\right) \bowtie_{P1.Addr = P2.Addr \wedge P1.SSN < P2.SSN} \left(\rho_{P2(\ldots)}\ Person\right)\right)\right)$$

still using SSN in join!

(my father & I shared the same name)

•
•
•

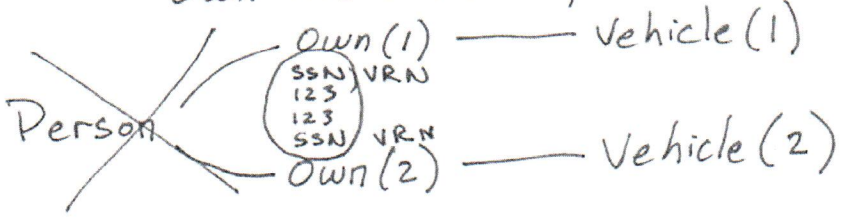
7. Write a query that lists pairs of Vehicles by Ma and Mo that are owned by the same Person.



$$\Pi_{V1.Ma,\ V1.Mo,\ V2.Ma,\ V2.Mo}\left(\left(\rho_{V1(..)}\text{Vehicle} \bowtie_{V1.VRN = O1.VRN} \rho_{O1(..)}\text{Own} \bowtie_{O1.SSN = Person.SSN} \text{Person} \bowtie_{Person.SSN = O2.SSN \wedge O2.VRN <> O1.VRN} \rho_{O2(..)}\text{Own} \bowtie_{O2.VRN = V2.VRN} \rho_{V2(..)}\text{Vehicle}\right)\right)$$

Can parenthize & rename in different ways

But bigger observation
~ don't need to involve person at all, because relation Own includes a person's unique identifier



$$\Pi_{V1.Ma,\ V1.Mo,\ V2.Ma,\ V2.Mo}\left(\left(\rho_{V1(..)}\text{Vehicle} \bowtie_{V1.VRN = O1.VRN} \rho_{O1(..)}\text{Own} \bowtie_{O1.SSN = O2.SSN \wedge O2.VRN <> O1.VRN} \rho_{O2(..)}\text{Own} \bowtie_{O2.VRN = V2.VRN} \rho_{V2(..)}\text{Vehicle}\right)\right)$$

would use < (or >) to get rid of conceptual redundancy

For problems like 7, or any problems of non-trivial complexity, its helpful to break them apart

① what relations are involved?

Ma, Mo requires that Vehicle be involved
if only VRNs were needed as the vehicle info then only own would be need

② decompose problem often helpful

simpler problem: list pairs of vehicles owned by same person

$$\mathcal{C}_{\substack{own \\ O1(..)}} \bowtie_{\substack{O1.SSN \\ = O2.SSN \\ \wedge O1.VRN \\ < O2.VRN}} \mathcal{C}_{\substack{own \\ O2(..)}}$$

then recognize that Ma, Mo required too, which implicates Vehicle

(or perhaps start with simpler problem of listing pairs of all different vehicles

$$\mathcal{C}_{\substack{Vehicle \\ V1(..)}} \bowtie_{\substack{V1.VRN \\ < \\ V2.VRN}} \mathcal{C}_{\substack{Vehicle \\ V2(..)}}$$

and build on that)

Notice that renaming in inner scope is OK to reference in outer scope (project). We will assume this to be OK for sake of ease of comprehension, even if some RA interpreters would flag an error. An alternative would rename globally (outermost scope, and reference inside that outermost scope: $\rho_{P1,\ldots}\rho_{P2}\rho_{O1}\rho_{O2}\rho_{V1}\rho_{V2}(\ldots)$

8. Write a query that lists pairs of Persons by SSN and Name that own a Vehicle of the same Ma, Mo and Color.

(but don't have to be same vehicle, & would not be in my interpretation, but in some settings the DB may allow for two owners of a vehicle — co-owners. The query below does not distinguish the two ~~conditions~~ possibilities

two different persons

$\Pi_{P1.SSN,\ P1.Name,\ P2.SSN,\ P2.Name}$ ($\rho_{P1(\ldots)}$ Person $\bowtie_{P1.SSN<P2.SSN}$ $\rho_{P2(\ldots)}$ Person

$\bowtie_{P1.SSN=O1.SSN}$ $\rho_{O1(\ldots)}$ Own $\bowtie_{P2.SSN=O2.SSN}$ $\rho_{O2(\ldots)}$ Own

each person owning (at least one) vehicle(s)

$\bowtie_{O1.VRN=V1.VRN}$ $\rho_{V1(\ldots)}$ Vehicle

$\bowtie_{O2.VRN=V2.VRN}$ $\rho_{V2(\ldots)}$ Vehicle

Vehicles owned by the different persons & have same Ma, Mo, & Color

$\wedge$ V1.Ma = V2.Ma

$\wedge$ V1.Mo = V2.Mo

$\wedge$ V1.Color = V2.Color

($\wedge$ V1.VRN < V2.VRN) — only if important to distinguish diff vehicles

9. Suppose that there was a Price attribute on Vehicle. Write a query in which each car is listed (paired) with each other car that costs less than it does.

$$\Pi_{V1.VRN,\ V2.VRN}\left(\rho_{V1(..)}\text{Vehicle} \bowtie_{V1.Price < V2.Price} \rho_{V2(..)}\text{Vehicle}\right)$$

Even if you didn't catch this before, this problem was to illustrate that not all joins are equality joins