CSC 370

Activity Worksheet: Relational Operators

Dr. Sean Chester

Fall 2022

Notes

This worksheet is intended to solidify your knowledge of relational operators by having you identify the (set of) operator(s) used to transform a given input relation into a given output relation. For each of the questions below, indicate what is the operator or sequence of operators that will transform the input into the output. Where more than one solution is possible, prefer the simplest. You will be even better prepared for subsequent material if you can express the operators using relational algebra. The first question provides an example of the expected solution.

Questions

1. Input Relation: Fruit - Output Relation: Fruit

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 1: Input Relation: Fruit

colour
blue
red
yellow

Table 2: Output Relation: Fruit

Solution: Project Fruit onto (colour). (Note: selection is not required because the extra 'yellow' tuple is already eliminated because of the duplicate elimination of sets.)

$$\pi_{\text{colour}}(\text{FRUIT})$$

2. Input Relation: Fruit - Output Relation: Fruit

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 3: Input Relation: Fruit

name	**colour**
banana	yellow
lemon	yellow

Table 4: Output Relation: Fruit

Solution:

3. Input Relation: Fruit - Output Relation: Fruit

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 5: Input Relation: Fruit

colour
yellow

Table 6: Output Relation: Fruit

Solution:

4. Input Relation: Fruit - Output Relation: YellowFruit

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 7: Input Relation: Fruit

name
banana
lemon

Table 8: Output Relation: YellowFruit

Solution:

5. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 9: Input Relation: Fruit

vegetable_name	**colour**
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 10: Input Relation: Vegetables

fruit_name	**colour**	**vegetable_name**
banana	yellow	corn
banana	yellow	squash
lemon	yellow	corn
lemon	yellow	squash
strawberry	red	capsicum

Table 11: Output Relation: Combine

Solution:

6. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 12: Input Relation: Fruit

vegetable_name	**colour**
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 13: Input Relation: Vegetables

fruit_name	**fruit.colour**	**vegetable.colour**	**vegetable_name**
banana	yellow	capsicum	red
banana	yellow	aubergine	purple
lemon	yellow	aubergine	purple
lemon	yellow	capsicum	red

Table 14: Output Relation: Combine

Solution:

7. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 15: Input Relation: Fruit

vegetable_name	**colour**
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 16: Input Relation: Vegetables

name	**colour**
banana	yellow
aubergine	purple
lemon	yellow
capsicum	red

Table 17: Output Relation: Combine

Solution:

8. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 18: Input Relation: Fruit

vegetable_name	**colour**
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 19: Input Relation: Vegetables

fruit_name	**fruit.colour**	**vegetable.colour**	**vegetable_name**
banana	yellow	aubergine	purple
banana	yellow	capsicum	red
blueberry	blue	squash	yellow
blueberry	blue	corn	yellow
blueberry	blue	aubergine	purple
blueberry	blue	capsicum	red
lemon	yellow	capsicum	red
lemon	yellow	aubergine	purple
strawberry	red	corn	yellow
strawberry	red	squash	yellow
strawberry	red	aubergine	purple

Table 20: Output Relation: Combine

Solution:

9. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	colour
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 21: Input Relation: Fruit

vegetable_name	colour
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 22: Input Relation: Vegetables

fruit_name	fruit.colour	vegetable.colour	vegetable_name
banana	yellow	aubergine	purple
banana	yellow	capsicum	red
lemon	yellow	capsicum	red
lemon	yellow	aubergine	purple
strawberry	red	corn	yellow
strawberry	red	squash	yellow
strawberry	red	aubergine	purple

Table 23: Output Relation: Combine

Solution:

10. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 24: Input Relation: Fruit

vegetable_name	**colour**
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 25: Input Relation: Vegetables

fruit_name	**vegetable_name**
banana	aubergine
banana	capsicum
lemon	capsicum
lemon	aubergine
strawberry	corn
strawberry	squash
strawberry	aubergine

Table 26: Output Relation: Combine

Solution:

11. Input Relation: Fruit and Input Relation: Vegetables - Output Relation: Combine

name	**colour**
blueberry	blue
banana	yellow
strawberry	red
lemon	yellow

Table 27: Input Relation: Fruit

vegetable_name	**colour**
aubergine	purple
corn	yellow
capsicum	red
squash	yellow

Table 28: Input Relation: Vegetables

dessert	**dinner**
banana	aubergine
lemon	aubergine
strawberry	aubergine

Table 29: Output Relation: Combine

Solution:

Solutions

Question 1

$$\pi_{\text{colour}}(\text{Fruit})$$

Selection is not required because the extra 'yellow' tuple is already eliminated because of the duplicate elimination of sets.

Question 2

$$\sigma_{\text{colour='yellow'}}(\text{Fruit})$$

We obtain the entire tuple in this case because there was no projection. We selected only those tuples for which the colour attribute had value 'yellow'.

Question 3

$$\sigma_{\text{colour='yellow'}}(\pi_{\text{colour}}(\text{Fruit}))$$

Observe that the order of these operators is interchangeable in this example.

Question 4

$$\rho_{\text{YellowFruit}}(\pi_{\text{name}}(\sigma_{\text{colour='yellow'}}))$$

Observe that this time the order of the projection and selection is not interchangeable. If you project first, then you will no longer have the attribute 'colour' with which do perform the section.

Question 5

$$\text{Fruit} \bowtie \text{Vegetable}$$

Here the only common attribute is 'colour'. So, a natural join will pair attributes from each relation that have the same colour. We get all 2×2 pairs of yellow fruits and yellow vegetables as well as the single pair of a red fruit and a red vegetable. Aubergine, and blueberry do not appear in the result at all because there are no purple fruits and no blue vegetables, respectively, in these tables.

Question 6

$$\text{Fruit} \bowtie_{\text{fruit.colour='yellow' AND vegetable.colour}\neq\text{'yellow'}} \text{Vegetable}$$

Again we need a join so that we can pair together rows from the fruit table with rows from the vegetable table. However, this result does not match fruits and vegetables with the same colour (the common attribute) so we need to define a custom condition (a theta join). Because there is a clash in name on a column that was not used as a join key ('colour'), each one is automatically renamed by prepending the table name for disambiguation. This is the same as the solution:

$$\sigma_{\text{colour='yellow'}}(\text{Fruit}) \times \sigma_{\text{colour}\neq\text{'yellow'}}(\text{Vegetable})$$

Question 7

$$\rho_{\text{Fruit(name,colour)}}(\sigma_{\text{colour='yellow'}}(\text{Fruit})) \cup \rho_{\text{Vegetable(name,colour)}}(\sigma_{\text{colour}\neq\text{'yellow'}}(\text{Vegetable}))$$

In this case, we do not want to match up pairs from each input relation, but just to select some results from each. Thus rather than the join, we need a classic set operator—union in this case. We take the results from Fruit that we want and union that result set with the vegetables that we want.

In order to use the set operators though, we need a common schema. Thus both relations have their leftmost attribute renamed.

Question 8

$$\text{Fruit} \bowtie_{\text{fruit.colour}\neq\text{vegetable.colour}} \text{Vegetable}$$

Observe another theta join in which we need to create a custom condition in order to retain the pairs of tuples that we want. In this case, we also set the condition to compare two values from different attributes rather than to compare one attribute to a constant literal. An equivalent solution is:

$$(\text{Fruit} \times \text{Vegetable}) \setminus (\text{Fruit} \bowtie \text{Vegetable})$$

Question 9

$$\sigma_{\text{name}\neq\text{'blueberry'}}(\text{Fruit}) \bowtie_{\text{fruit.colour}\neq\text{vegetable.colour}} \text{Vegetable}$$

This question is the same as the previous one, except that we need to filter out some results. This can be done by applying a selection before the join.

Question 10

$$\pi_{\text{Result(fruit\_name, vegetable\_name)}}(\sigma_{\text{name}\neq\text{'blue'}}(\text{Fruit}) \bowtie_{\text{fruit.colour}\neq\text{vegetable.colour}} \text{Vegetable})$$

As above, except that we need to project away the colour attributes. Note that this projection must not be done prior to the join because the colour attribute is used as part of the theta-join predicate.

Question 11

$$\rho_{(\text{dessert, dinner})}(\pi_{(\text{fruit\_name, vegetable\_name})}(\sigma_{\text{name}\neq\text{'blue'}}(\text{Fruit}) \times \sigma_{\text{vegetable\_name}=\text{'aubergine'}}(\text{Vegetable}))$$

As above, except that we need to project away the colour attributes. Note that this projection must not be done prior to the join because the colour attribute is used as part of the theta-join predicate.