CS 143
January 17, 2021

Homework 1

Jason Lai
204995126
Dis 1B

1 Problem 1

$(R-S) \cup (S-R)$

A	B	C
7	5	3
1	4	3
6	7	9
1	4	4
8	3	2

Work:

R

A	B	C
7	5	3
2	1	2
1	4	3
5	8	7
6	7	9

S

A	B	C
2	1	2
1	4	4
8	3	2
5	8	7

$R - S$

A	B	C
7	5	3
1	4	3
6	7	9

$S - R$

A	B	C
1	4	4
8	3	2

2 Problem 2

$\sigma_{R.L > S.M \wedge R.M < S.P}(R \times S)$

R.L	R.M	S.M	S.N	S.P
4	3	1	6	4
4	3	3	4	7
6	5	3	4	7
8	7	6	1	8

Work:

R

L	M
4	3
6	5
8	7

S

M	N	P
6	1	8
1	6	4
2	5	1
3	4	7

$R \times S$

R.L	R.M	S.M	S.N	S.P
4	3	6	1	8
4	3	1	6	4
4	3	2	5	1
4	3	3	4	7
6	5	6	1	8
6	5	1	6	4
6	5	2	5	1
6	5	3	4	7
8	7	6	1	8
8	7	1	6	4
8	7	2	5	1
8	7	3	4	7

3 Problem 3

a) $(\pi_{Student-name}(Student))-(\pi_{Student-name}(\sigma_{Course-name=\text{'Database Management Systems'}}(Enrollment)))$

b) $\pi_{Student.Student-name}(\sigma_{Student.Department<>Course.Department}(\sigma_{Student.Student-name=Enrollment.Student-name}(Student \times (Enrollment \bowtie Course))))$

c) $(\pi_{Course-name}(Course)) - (\pi_{Course-name}(Enrollment))$

d) $\pi_{Student.Department}(\sigma_{Student.Student-name=Enrollment.Student-name}(Student\times(\sigma_{Department=\text{'CS'}}(Enrollment \bowtie Course))))$

e) $\pi_{Department}(Student-(\pi_{Student-name,Department}(Student \bowtie (\sigma_{Enrollment.Course-name<>E2.Course-name}(\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollment \times \rho_{E2}(Enrollment)))))))$

Work:

a)
Use the $Enrollment$ relation:

$\quad Enrollment$

Select only the tuples that have the 'Database Management Systems' $Course-name$:

$\quad \sigma_{Course-name=\text{'Database Management Systems'}}(Enrollment)$

Filter out the $Course-name$ column, so that we only have the names of the students:

$\quad \pi_{Student-name}(\sigma_{Course-name=\text{'Database Management Systems'}}(Enrollment))$

We want the names of all students who are not in 'Database Management Systems', so we use the $Student$ relation:

$\quad Student$

Filter out the $Department$ column, so that we only have the names of the students:

$\quad \pi_{Student-name}(Student)$

We now use the difference operator to get the names of all the students in the $Student$ relation that are not in the relation we have above, which are the names of all the student enrolled in 'Database Management Systems':

$\quad (\pi_{Student-name}(Student))-(\pi_{Student-name}(\sigma_{Course-name=\text{'Database Management Systems'}}(Enrollment)))$

b)
Use the $Enrollment$ relation:

$\quad Enrollment$

Natural join with the $Course$ relation, so we get the courses students are enrolled in with the departments those courses belong to:

$Enrollment \bowtie Course$

Cross product the *Student* relation and the relation above:

$Student \times (Enrollment \bowtie Course)$

Select only the tuples with matching student names:

$\sigma_{Student.Student-name=Enrollment.Student-name}(Student \times (Enrollment \bowtie Course))$

Select only the tuples with mismatched departments:

$\sigma_{Student.Department<>Course.Department}(\sigma_{Student.Student-name=Enrollment.Student-name}(Student \times (Enrollment \bowtie Course)))$

Filter out all columns except the students' names:

$\pi_{Student.Student-name}(\sigma_{Student.Department<>Course.Department}(\sigma_{Student.Student-name=Enrollment.Student-name}(Student \times (Enrollment \bowtie Course))))$

c)

Use the *Enrollment* relation:

$Enrollment$

Filter out the $Student - name$ column, so that we only have the names of the courses that have students enrolled:

$\pi_{Course-name}(Enrollment)$

We want the names of all courses which have no students enrolled, so we use the *Course* relation:

$Course$

Filter out the *Department* column, so that we only have the names of the courses:

$\pi_{Course-name}(Course)$

We now use the difference operator to get the names of all of the courses in the *Course* relation that are not in the relation we have above, which are names of all the courses with students enrolled:

$(\pi_{Course-name}(Course)) - (\pi_{Course-name}(Enrollment))$

d)

Use the *Enrollment* relation:

$Enrollment$

Natural join with the *Course* relation, so we get the courses students are enrolled in with the departments those courses belong to:

$Enrollment \bowtie Course$

Select only the students who are taking courses that belong to the CS department:

$\sigma_{Department=\text{'CS'}}(Enrollment \bowtie Course)$

We want the departments of all the students who are taking at least one class offered by the CS department, so we use the *Student* relation:

$Student$

We then cross product the *Student* relation and the relation we found above:

$$Student \times (\sigma_{Department=\text{'CS'}}(Enrollment \bowtie Course))$$

Select only the tuples with a matching Student.Student-name and Enrollment.Student-name, so we only get the tuples of students in a CS course:

$$\sigma_{Student.Student-name=Enrollment.Student-name}(Student \times (\sigma_{Department=\text{'CS'}}(Enrollment \bowtie Course)))$$

Filter out all other columns besides the students' departments:

$$\pi_{Student.Department}(\sigma_{Student.Student-name=Enrollment.Student-name}(Student \times (\sigma_{Department=\text{'CS'}}(Enrollment \bowtie Course))))$$

e)

Use the *Enrollment* relation:

$$Enrollment$$

We the self join the *Enrollment* relation:

$$Enrollment \times \rho_{E2}(Enrollment)$$

Select the tuples with matching student names:

$$\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollment \times \rho_{E2}(Enrollment))$$

Select the tuples with mismatched course names, as this means a student is enrolled in 2 or more courses:

$$\sigma_{Enrollment.Course-name<>E2.Course-name}(\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollment \times \rho_{E2}(Enrollment)))$$

Natural join with the *Student* relation to concatenate a *Department* attribute for the students that are enrolled in 2 or more courses:

$$Student \bowtie (\sigma_{Enrollment.Course-name<>E2.Course-name}(\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollmen \times \rho_{E2}(Enrollment))))$$

Natural join combined all 3 of the $Student-name$ attributes into one, so now we remove the $Course-name$ columns, so that our result has the same schema as the *Student* relation:

$$\pi_{Student-name,Department}(Student \bowtie (\sigma_{Enrollment.Course-name<>E2.Course-name}(\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollment \times \rho_{E2}(Enrollment)))))$$

Now we find the difference between the *Student* relation and the result thus far to get the tuples of students who are not taking more than one class:

$$Student - (\pi_{Student-name,Department}(Student \bowtie (\sigma_{Enrollment.Course-name<>E2.Course-name}(\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollment \times \rho_{E2}(Enrollment))))))$$

Finally, we filter out the student names to get just the department names:

$$\pi_{Department}(Student-(\pi_{Student-name,Department}(Student \bowtie (\sigma_{Enrollment.Course-name<>E2.Course-name}(\sigma_{Enrollment.Student-name=E2.Student-name}(Enrollment \times \rho_{E2}(Enrollment)))))))$$

4 Problem 4

A relational algebra expression to find the names of the lowest valued companies is:

$\pi_{company-name}(Company-(\pi_{Company.company-name,Company.valuation}(\sigma_{Company.valuation>C2.valuation}(Company\times \rho_{C2}(Company)))))$

Work:

First, we self join the *Company* relation:

$Company \times \rho_{C2}(Company)$

Then, we select all of the tuples where $Company.valuation > C2.valuation$, to get the $Company.company - name$ names that do not have the lowest valuation. This follows the logic that if a company has a valuation greater than at least one other company, then it does not have the lowest valuation:

$\sigma_{Company.valuation>C2.valuation}(Company \times \rho_{C2}(Company))$

We then filter out the $C2$ attributes, to get a schema that is the same as the *Company* relation:

$\pi_{Company.company-name,Company.valuation}(\sigma_{Company.valuation>C2.valuation}(Company\times\rho_{C2}(Company)))$

We then find the difference between the *Company* relation and our result thus far to get the tuples of companies with the lowest valuation:

$Company-(\pi_{Company.company-name,Company.valuation}(\sigma_{Company.valuation>C2.valuation}(Company \times \rho_{C2}(Company))))$

We then filter out the *valuation* attribute from the result to get just the names of the companies with the lowest valuation:

$\pi_{company-name}(Company-(\pi_{Company.company-name,Company.valuation}(\sigma_{Company.valuation>C2.valuation}(Company \times \rho_{C2}(Company)))))$