# Solution to the Assignment 2

Dirk Van Gucht & Mo Zhou

September 29, 2008

## Relational Algebra

- We write r(ABC) for a relational schema with relation name r and attributes (or fields) A, B, and C. The underlined attribute is the primary key.

## Basic Algebra

1. Given following relational schema, write expressions of relational algebra to answer the following queries.

   Relational schema of question 1.
   Product(model, maker, type)
   PC(model, speed, ram, hd, price)
   Laptop(model, speed, ram, hd, screen, price)
   Printer(model, color, type, price)

The sample data for relations of question 1. These data are not used to calculate results of following questions.

Product:

| model | maker | type |
| --- | --- | --- |
| 1001 | A | PC |
| 3001 | B | Printer |
| 2001 | C | laptop |

PC:

| model | speed | ram | hd | price |
| --- | --- | --- | --- | --- |
| 1001 | 2.66 | 1024 | 250 | 2114 |
| 1002 | 1.42 | 512 | 250 | 955 |
| 1003 | 3.20 | 2048 | 160 | 1049 |

Laptop:

| model | speed | ram | hd | screen | price |
| --- | --- | --- | --- | --- | --- |
| 2001 | 2.00 | 1024 | 250 | 15 | 2114 |
| 2002 | 1.73 | 512 | 80 | 24 | 955 |
| 2003 | 1.83 | 2048 | 60 | 20 | 1049 |

Printer:

| model | color | type | price |
| --- | --- | --- | --- |
| 3001 | true | ink-jet | 99 |
| 3002 | false | laser | 239 |
| 3003 | true | laser | 899 |

(a) What PC models have a speed of at least 3.00?
$\pi_{model}(\sigma_{model \geq 3.00}(PC))$

(b) Find the model numbers of all color laser printers.
$\pi_{model}(\sigma_{type=laser \wedge color=true}(Printer))$

(c) Which manufacturers make laptops with a hard disk of at least 100GB?
$\pi_{maker}(Product \bowtie \sigma_{hd \geq 100}(Laptop))$

(d) Find the model number and price of all products (of any type) made by manufacturer B.
$\pi_{model,price}(\sigma_{maker=B}(Product)) \bowtie$
$(\pi_{model,price}(PC) \cup \pi_{model,price}(Laptop) \cup \pi_{model,price}(Printer))$

(e) Find those manufacturers that sell Laptops, but not PC's.
$\pi_{maker}(\sigma_{type=Laptop}(Product)) - \pi_{maker}(\sigma_{type=PC}(Product))$

(f) Find those manufacturers that sell all models of PCs and lazer Printers.
$\pi_{maker}(Product/(\pi_{model}(PC) \cup \pi_{model}(\sigma_{type=lazer}(Printer))))$

(g) Find those manufacturers whose laptops have all ram sizes that manufacturer B's laptops have.
$(\pi_{maker,ram}(Product \bowtie Laptop))$ / $(\pi_{ram}(\sigma_{maker=B}(Product) \bowtie Laptop))$

(h) Find those manufacturers of at least two different computers (PC's or laptops) with speeds of at least 2.80.
$\rho(TEMP1, \pi_{model}(\sigma_{speed \geq 2.80}(PC)) \cup$
$\pi_{model}(\sigma_{speed \geq 2.80}(Laptop)))$
$\rho(TEMP2, Product \bowtie TEMP1)$
$\pi_{maker}(\sigma_{TEMP2.maker=TEMP3.maker \wedge TEMP2.model \neq TEMP3.model}(TEMP2 \times$
$\rho(TEMP3, TEMP2)))$

(i) Find the manufacturers of PC's with at least two different speeds.
$\pi_{R1.maker}(\sigma_{R1.speed \neq R2.speed \wedge R1.maker=R2.maker}((\rho(R1, (Product \bowtie$
$PC)) \times \rho(R2, (Product \bowtie PC)))))$

(j) Find the manufacturers who sell exactly two different models of PC.
$\pi_{R1.maker}(\sigma_{R1.model \neq R2.model \wedge R1.maker=R2.maker}(\rho(R1, (Product \bowtie$
$PC)) \times \rho(R2, (Product \bowtie PC)))) -$
$\pi_{R1.maker}(\sigma_{R1.model \neq R2.model \wedge R1.model \neq R3.model \wedge R2.model \neq R3.model}$
$_{\wedge R1.maker=R2.maker \wedge R1.maker=R3.maker}$

$(\rho(R1, (Product \bowtie PC)) \times \rho(R2, (Product \bowtie PC)) \times \rho(R3, (Product \bowtie PC))))$

(k) Find those pairs of PC models that have both the same speed and RAM. A pair should be listed only once; e.g., list (i, j) but not (j, i).
*Hint: The model numbers can be compared.*

$\pi_{PC1.model, PC2.model}$
$(\sigma_{PC1.speed=PC2.speed \wedge PC1.ram=PC2.ram \wedge PC1.model>PC2.model}$
$(\rho(PC1, PC) \times \rho(PC2, PC)))$

(l) Find the manufacturer(s) of the computer (PC or laptop) with the highest available speed.
*Hint: the highest speed means that it is not smaller than any other speeds. If you can find all the speeds which are smaller than some speed, you can solve this problem.*
$\rho(Computer, (\pi_{model,speed}(PC) \cup \pi_{model,speed}(Laptop)))$

$\rho(HighestSpeedModel, \pi_{model}(Computer) - \pi_{model}(\sigma_{P1.speed<P2.speed}(\rho(P1, Computer) \times \rho(P2, Computer))))$

$\pi_{maker}(Product \bowtie HighestSpeedModel)$

## Constraints

2. Express the following constraints about the relations below.
   Product(<u>model</u>,maker, type)
   PC(<u>model</u>, speed, ram, hd, price)
   Laptop(<u>model</u>, speed, ram, hd, screen, price)
   Printer(<u>model</u>, color, type, price)

   (a) A PC with a processor speed less than 2.00 must not sell for more than \$500.
   $\sigma_{speed<2.00 \wedge price>500}(PC) = \emptyset$

   (b) No manufacturer of PC's may also make laptops.
   $\pi_{maker}(Product \bowtie PC) \cap \pi_{maker}(Product \bowtie Laptop) = \emptyset$
   OR
   $\pi_{maker}(\sigma_{type=PC}(Product)) \cap \pi_{maker}(\sigma_{type=Laptop}(Product)) = \emptyset$

   (c) A manufacturer of a PC must also make a laptop with at least as great a processor speed.
   Find the highest speed of PC for each manufacturer.

   $\rho(Temp, \pi_{maker,speed}(Product \bowtie PC))$
   $\rho(Temp2, Temp -$
   $\pi_{T1.maker,T1.speed}(\sigma_{T1.speed<T2.speed \wedge T1.maker=T2.maker}(\rho(T1, Temp) \times$
   $\rho(T2, Temp))))$

   Find the highest speed of Laptop for each manufacturer

   $\rho(Temp3, \pi_{maker,speed}(Product \bowtie Laptop))$
   $\rho(Temp4, Temp3 -$
   $\pi_{T3.maker,T3.speed}(\sigma_{T3.speed<T4.speed \wedge T3.maker=T4.maker}(\rho(T3, Temp) \times$
   $\rho(T4, Temp))))$

   For the same manufacturer, the highest speed of the laptop must as great as the one of the PC.

   $\sigma_{Temp2.maker=Temp4.maker \wedge Temp2.speed \geq Temp4.speed}(Temp2 \times$
   $Temp4) = \emptyset$

3. Express the following constraints in relational algebra. The constraints are based on the following relations.
Classes(class, type, country, numGuns, bore, displacement)
Ships(shipname, class, launched)
Battles(battlename, date)
Outcomes(shipname, battlename, result)

   (a) No country may have both battleships and battlecruisers.(battleship and battlecruiser are values of attribute type in relation Classes)
   $\pi_{contry}(\sigma_{type=battlecruiser}(Classes)) \cap$
   $\pi_{contry}(\sigma_{type=battleship}(Classes)) = \emptyset$
   (b) No ship with more than 9 guns may be in a battle with a ship having fewer than 9 guns that was sunk. (sunk is a value of attribute result in relation Outcomes)
   $\pi_{battlename}((\sigma_{numGuns>9}(Classes) \bowtie Ships) \bowtie Battles)\cap$
   $\pi_{battlename}((\sigma_{numGuns<9}(Classes) \bowtie Ships) \bowtie$
   $(\sigma_{result=sunk}(Battles))) = \emptyset$

## Special algebra

4. Suppose relations R and S have n tuples and m tuples, respectively. Give the minimum and the maximum numbers of tuples that the results of the following expressions can have.

   (a) $R \cup S$

   (b) $R \bowtie S$

   (c) $\sigma_C(R) \times S$, for some condition C.

   (d) $\pi_L(R) - S$, for some list of attributes L.

| Expression | Max | Min |
|---|---|---|
| $R \cup S$ | n+m | max(m,n) |
| $R \bowtie S$ | min(m,n) | 0 |
| $\sigma_C(R) \times S$ | $m \times n$ | 0 |
| $\pi_L(R) - S$ | n | 0 |

5. The *semijoin* of relations R and S is the set of tuples t in R such that there is at least one tuple in S that agrees with t in all attributes that R and S have in common. Give three different expressions of relational algebra equivalent to *semijoin*.
   Suppose that R has n domains and S has m domains. And the first k domains are the same for both R and S.

   (a) $\pi_{1,\ldots,n}(R \bowtie S)$

   (b) $\pi_{1,\ldots,n}(\sigma_{R.1=S.1 \wedge \ldots \wedge R.k=S.k}(R \times S))$

   (c) $R \bowtie \pi_{1,\ldots,k}(S)$

6. The *antisemijoin* is the set of tuples t in R that do not agree with any tuple of S in the attributes common to R and S. Give an expression of relational equivalent to *antisemijoin*.
   $R \ - \ \pi_{1,\ldots,n}(R \bowtie S)$

7. An operator on relations is said to be *monotone* if whenever we add a tuple to one of its arguments, the result contains all the tuples that it contained before adding the tuple, plus perhaps more tuples. Are semijoin and antisemijoin monotone? For each, either explain why it is monotone or give an example showing it is not.

Solution: Semijoin is monotone.
Antisemijoin is not monotone.