COS 597A, Fall 2011
Solutions to Problem Set 2

1.

Relation R	Item	Size	Color	Finish
	table	lg	white	antique
	table	lg	white	glossy
	table	sm	white	glossy
	table	lg	red	glossy
	table	sm	red	glossy
	chair	lg	white	antique
	chair	lg	white	glossy
	chair	lg	red	glossy

Relation Q	Color	Finish
	white	antique
	white	glossy
	red	glossy

Relation R÷Q	Item	Size
	table	lg
	chair	lg

Relation (R÷Q) X Q	Item	Size	Color	Finish
	table	lg	white	antique
	table	lg	white	glossy
	table	lg	red	glossy
	chair	lg	white	antique
	chair	lg	white	glossy
	chair	lg	red	glossy

(table, sm, white, glossy) is not in (R÷Q) X Q.

2.
Part a.
{a} and {b, c} are candidate keys for R − Q. {b} is a foreign key referencing X.x.

Part b.

{a} is a candidate key for R÷T and {b} is a foreign key referencing X.x. Division is defined if T is the empty set. However it is interesting to note that if we require T to be not empty to perform division, then we can conclude {b} is a candidate key for R÷T. To see this, suppose {b} is not a candidate key for R÷T. Then, for some value b1 in B and distinct values a1 and a2 in A, one can have (a1, b1) and (a2, b1) both in R÷T. But then by the definition of the division operation in relational algebry, for any tuple (c1, d1) in T, tuples (a1,b1,c1,d1) and (a2,b1,c1,d1) must be in R. However {b, c} is a candidate key in R, which means a1and a2 cannot be distinct (there can be only one tuple in R with a given pair of values for b and c. Hence we derive a contradiction if we assume {b} is not a candiate key for R÷T.

3. i $\pi_{name}(\sigma_{co\_name='Microsoft' \wedge salary \leq 30000}$ works)

3.ii $\pi_{name,co\_name}(\sigma_{city='Trenton' \wedge salary>1000000}$ (employee ⊳⊲ works))

3.iii π_{co_name} (company) - π_{co_name} ($\sigma_{city='Princeton'}$ (company))

3.iv company ÷ (π_{city}($\sigma_{co\ name='Fred's\ Pizza\ Co.'}$ (company))

3.v employee ⊳⊲ (

($\pi_{name,\ name2}$

($\sigma_{salary > salary2}$

(works X $\rho_{name\rightarrow name2,\ co\_name\rightarrow co\_name2,\ salary\rightarrow salary2}$ (works))))

÷

(π_{name2}

($\sigma_{co\_name='IBM'}$

($\rho_{manager\_name \rightarrow name2}$ (manages) ⊳⊲ $\rho_{name \rightarrow name2}$ (works))))

)

4.i {T | ∃S ε works (S.co name = 'Microsoft' ∧ S.salary ≤ 30000 ∧
T[name] = S[name])}

4.ii {T | ∃E ε employee ∃W ε works
(E[name] = W[name] ∧ E[city] = 'Trenton' ∧ W[salary] > 1000000
∧ T[name] = E[name] ∧ T[co_name] = W[co_name]}

4.iii {T | ∃C_1 ε company((T[co_name] = C_1[co_name]) ∧
(∀C_2 ε company (C_2[co_name] ≠ C_1[co name]
∨ C_2[city] ≠ 'Princeton'))
) }

4.iv $\{T \mid \exists C_1 \, \varepsilon \text{ company } (C_1[co\_name] = T[co\_name] \wedge$
$(\forall C_2 \, \varepsilon \text{ company } ((C_2[co\ name] = \text{'Fred'sPizzaCo.'})$
$\Rightarrow (\exists C_3 \, \varepsilon \text{ company } (C_3[co\_name] = C_1[co\_name]$
$\wedge C_3[city] = C_2[city]) \quad)))) \quad \}$

4.v $\{T \mid T \, \varepsilon \text{ employee } \wedge$
$\exists W_1 \, \varepsilon \text{ works } (T[name] = W_1[name] \wedge$
$(\forall M \, \varepsilon \text{ manages } \forall W_2 \, \varepsilon \text{ works}$
$(\quad (M[manager\_name] = W_2[name] \wedge W_2[co\ name] = \text{'IBM'})$
$\Rightarrow W_1[salary] > W_2[salary] \quad) \quad) \quad) \quad \}$

5.

Part a

Two common alternatives:

i. $\pi_{attr1}\pi_{attr2}\ldots\pi_{attrk}(rel_1 X\ rel_2 X \ldots X\ rel_k)$

ii. $\pi_{attr1}(rel_1)\ X\ \pi_{attr2}(rel_2) X \ldots X\ \pi_{attrk}(rel_k)$

In alternative i, the cross-product establishes the existence of the tuples of rel_1 through rel_k, giving all possible choices of combinations of tuples, analogous to the "$\exists R_1 \ldots \exists R_k(R_1 \varepsilon\ rel_1 \wedge \ldots \wedge R_k \varepsilon\ rel_k$" allowing an arbitrary choice of k tuples, one from each rel_i. The projection chooses the attributes of the resulting tuples.

In alternative ii, the role of projection in capturing $\exists$ is more obvious: $\pi_{attri}(rel_i)$ says we are projecting away unneeded attributes of a tuple that exists in rel_i.

Part b

Let rel_1 have attributes a_1 through a_p and rel_2 have attributes b_1 through b_p. Then the equivalent tupe relational calculus query for $rel_1 X rel_2$ is

$$\{T \mid \exists R_1 \exists R_2(R_1 \varepsilon\ rel_1 \wedge R_2 \varepsilon\ rel_2 \wedge T[attr_1]=R_1[a_1] \wedge T[attr_2]=R_1[a_2] \wedge \cdots \wedge T[attr_p]=R_1[a_p] \wedge T[attr_{p+1}]=R_2[b_1] \wedge T[attr_{p+2}]=R_2[b_2] \wedge \cdots \wedge T[attr_{p+q}]=R_2[b_q]\}$$