# Fall, 2003    CIS 550

## Database and Information Systems

## Solutions to Homework 1

**Problem 1**: Consider the Penn Ebay (PBAY) System which is represented by the following schema:

Sellers(*sellerID*:integer,*rating*:char,*email*:string)
Items(*itemID*:integer,*description*:string,*startBid*:real,*sellerID*:integer,*qty*:integer)
Purchases(*purchaseNumber*:integer,*itemID*:integer,*custID*:integer,*count*:integer,*soldFor*:real)
Customers(*custID*:integer,*address*:string)

Write the following queries in relational algebra, tuple relational calculus and domain relational calculus:

*Note that **DRC** is similar to **TRC** except that we explicitely write the entries in the tuple. For example,* $\exists I \in Item$ *will be written as* $\exists iid, desc, sbid, sid, q, < iid, desc, sbid, sid, q > \in Items$ *and instead of checking for something like* $I.startBid > 1000$ *we will check for* $sbid > 1000$*. The logical structure remains the same. Hence, we'll only provide TRC here.*

1. Find the ID's of sellers of items with starting bid $\geq$ \$1000
   **RA**: $\pi_{sellerID}(\sigma_{startBid \geq 1000} Items)$
   **TRC**:$\{R | \exists I \in Items(I.startBid \geq 1000 \wedge R.sellerID = I.sellerID)\}$

2. Find the ID's of customers who bought $\geq 2$ of the same item or bought an item that a seller had with quantity 1.
   **RA**:$\pi_{custID}(\sigma_{count \geq 2} Purchase) \cup \pi_{custID}(\sigma_{qty=1}(Items \bowtie Purchases))$
   **TRC**:$\{R | \exists P \in Purchases((P.count \geq 2 \vee (\exists I \in Items(P.itemID = I.itemID \wedge I.qty = 1)) \wedge (R.custID = P.custID))\}$

3. Find the ID's of items stocked by every seller with rating A
   **RA**:$\pi_{itemID,sellerID}(Items) / \pi_{sellerID}(\sigma_{rating='A'} Sellers)$
   **TRC**:$\{R | \exists I \in Items((\forall S \in Sellers(S.rating =' A' \Rightarrow (I.sellerID = S.sellerID))) \wedge (R.itemID = I.itemID))\}$

4. Find the ID's of items which are stocked by $\geq 2$ sellers.
   **RA**: $\rho(Items2(itemID \rightarrow itemID2, sellerID \rightarrow sellerID2), Items)$
   $\pi_{itemID}(\sigma_{itemID=itemID2 \vee sellerID \neq sellerID2}(Items \times Items2))$
   or
   $\rho(Items2(itemID \rightarrow itemID2, sellerID \rightarrow sellerID2), Items)$
   $\pi_{itemID}(Items \bowtie_{itemID=itemID2 \vee sellerID \neq sellerID2} Items2)$
   **TRC**:$\{R | \exists I_1, I_2 \in Items(I_1.itemID = I2.itemID \wedge I_1.sellerID \neq I2.sellerID \wedge R.itemID = I1.itemID\}$

5. Find the ID's of items which are stocked by $\geq 2$ sellers who have different starting bids for the item.
   This part is similar to part 4, except that we need to check for an extra condition, $startBid \neq startBid2$

6. Find the ID's of items that are only sold for $\leq \$1000$, by any seller.
   **RA**:$\pi_{itemID}(Purchases) - \pi_{itemID}(\sigma_{soldFor>1000}Purchases)$
   **TRC**:$\{R|\neg(\exists P \in Purchases(P.soldFor > 1000) \wedge (P.itemID = R.itemID))\}$

**Problem 2**: Consider the following schema:

Suppliers(*sid*:integer,*sname*:string,*address*:string)
Parts(*pid*:integer,*pname*:string,*color*:string)
Catalog(*sid*:integer,*pid*:integer,*cost*:real)

State what the following queries compute:

1. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers))$
   Invalid query.

2. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog))) \bowtie Suppliers)$
   Names of suppliers who supply a red part costing less than \$100.

3. $(\pi_{sname}((\sigma_{color='red'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)) \cap$
   $(\pi_{sname}((\sigma_{color='green'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers))$
   Names of suppliers, where at least one of the suppliers with that name supplies a red part for less than \$100 and at least one of the suppliers with that name supplies a green part for less than \$100.

4. $(\pi_{sid}((\sigma_{color='red'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)) \cup$
   $(\pi_{sid}((\sigma_{color='green'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers))$
   IDs of suppliers supplying a red part at less than \$100 or a green part for less than \$100

5. $\pi_{sname}((\pi_{sid,sname}((\sigma_{color='red'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)) \cap$
   $(\pi_{sid,sname}((\sigma_{color='green'}(Parts)) \bowtie (\sigma_{cost<100}(Catalog)) \bowtie Suppliers)))$
   Names of suppliers who supply a red part and a green part each of which cost less than \$100.

**Problem 3**: Problem 4.6 from the textbook. It is reproduced here.
What is *relational completeness*? If a query language is relationally complete, can you write any desired query in that language?
*Relational completeness* means that a query language can express every query that can be written in relational algebra. It does not mean that the language can express any given query (for example, aggregation, recursion, etc.).