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 explicitly write the entries in the tuple. For example, \(\exists i \in I\) will be written as \(\exists iid, desc, sbid, sid, q < iid, desc, sbid, sid, q > \in I\) 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:** \(\{R \mid \exists I \in Items(I: startBid \geq 1000 \land R: sellerID = I: sellerID)\}\)

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:** \(\pi_{custID}(\sigma_{count \geq 2} Purchase) \cup \pi_{custID}(\sigma_{qty = 1} (Items \land Purchases))\)

   **TRC:** \(\{R \mid \exists P \in Purchases(P: count \geq 2 \lor (\exists I \in Items(I: itemID = P: itemID \land I: qty = 1)) \land (R: custID = P: custID)\}\)

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:** \(\{R \mid \exists I \in Items(\forall S \in Sellers(S: rating = 'A' \Rightarrow (I: sellerID = S: sellerID))) \land (R: itemID = I: itemID)\}\)

4. Find the ID’s of items which are stocked by \(\geq 2\) sellers.

   **RA:** \(\rho(Items2(itemID \rightarrow itemID2, sellerID \rightarrow sellerID2), Items)\)

   \(\pi_{itemID}(\sigma_{itemID = itemID2 \land sellerID = sellerID2} (Items \times Items2))\)

   or

   \(\rho(Items2(itemID \rightarrow itemID2, sellerID \rightarrow sellerID2), Items)\)

   \(\pi_{itemID}(Items \land itemID = itemID2 \land sellerID = sellerID2 Items2)\)

   **TRC:** \(\{R \mid \exists I_1, I_2 \in Items(I_1: itemID = I_2: itemID \land I_1: sellerID \neq I_2: sellerID \land R: itemID = I_1: itemID)\}\)
5. Find the ID’s of items which are stocked by ≥ 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, \( \text{startBid} \neq \text{startBid2} \)

6. Find the ID’s of items that are only sold for ≤ $1000, by any seller.

RA: \( \pi_{\text{itemID}}(\text{Purchases}) - \pi_{\text{itemID}}(\sigma_{\text{soldFor} > 1000}(\text{Purchases})) \)

TRC: \( R \models (\exists P \in \text{Purchases}(P.\text{soldFor} > 1000) \land (P.\text{itemID} = R.\text{itemID})) \)

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

- Suppliers(\( \text{sid}: \text{integer}, \text{sname}: \text{string}, \text{address}: \text{string} \))
- Parts(\( \text{pid}: \text{integer}, \text{pname}: \text{string}, \text{color}: \text{string} \))
- Catalog(\( \text{sid}: \text{integer}, \text{pid}: \text{integer}, \text{cost}: \text{real} \))

State what the following queries compute:

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

   Invalid query.

2. \( \pi_{\text{sname}}(\pi_{\text{sid}}((\sigma_{\text{color}='\text{red}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers}) \)

   Names of suppliers who supply a red part costing less than $100.

3. \( (\pi_{\text{sname}}((\sigma_{\text{color}='\text{red}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers})) \land \\
\( (\pi_{\text{sname}}((\sigma_{\text{color}='\text{green}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers})) \)

   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. \( (\pi_{\text{sid}}((\sigma_{\text{color}='\text{red}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers})) \cup \\
(\pi_{\text{sid}}((\sigma_{\text{color}='\text{green}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers})) \)

   IDs of suppliers supplying a red part at less than $100 or a green part for less than $100.

5. \( \pi_{\text{sname}}((\pi_{\text{sid,sname}}((\sigma_{\text{color}='\text{red}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers})) \land \\
(\pi_{\text{sid,sname}}((\sigma_{\text{color}='\text{green}'(\text{Parts})}) \bowtie (\sigma_{\text{cost}<100}(\text{Catalog})) \bowtie \text{Suppliers})) \)

   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.).