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

\[
\begin{align*}
\text{Sellers}(\text{sellerID}: \text{integer}, \text{rating}: \text{char}, \text{email}: \text{string}) \\
\text{Items}(\text{itemID}: \text{integer}, \text{description}: \text{string}, \text{startBid}: \text{real}, \text{sellerID}: \text{integer}, \text{qty}: \text{integer}) \\
\text{Purchases}(\text{purchaseNumber}: \text{integer}, \text{itemID}: \text{integer}, \text{custID}: \text{integer}, \text{count}: \text{integer}, \text{soldFor}: \text{real}) \\
\text{Customers}(\text{custID}: \text{integer}, \text{address}: \text{string})
\end{align*}
\]

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

1. Find the ID’s of sellers of items with starting bid $\geq 1000$
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.
3. Find the ID’s of items stocked by every seller with rating A
4. Find the ID’s of items which are stocked by $\geq 2$ sellers.
5. Find the ID’s of items which are stocked by $\geq 2$ sellers who have different starting bids for the item.
6. Find the ID’s of items that are only sold for $\leq 1000$, by any seller.

Problem 2: Consider the following schema:

\[
\begin{align*}
\text{Suppliers}(\text{sid}: \text{integer}, \text{sname}: \text{string}, \text{address}: \text{string}) \\
\text{Parts}(\text{pid}: \text{integer}, \text{pname}: \text{string}, \text{color}: \text{string}) \\
\text{Catalog}(\text{sid}: \text{integer}, \text{pid}: \text{integer}, \text{cost}: \text{real})
\end{align*}
\]

State what the following queries compute:

1. $\pi_{\text{sname}}(\pi_{\text{sid}}(\sigma_{\text{color}=\text{red}}(\text{Parts})) \Join (\sigma_{\text{cost}<100}(\text{Catalog})) \Join \text{Suppliers})$
2. $\pi_{\text{sname}}(\pi_{\text{sid}}((\sigma_{\text{color}=\text{red}}(\text{Parts})) \Join (\sigma_{\text{cost}<100}(\text{Catalog}))) \Join \text{Suppliers})$
3. $(\pi_{\text{sname}}((\sigma_{\text{color}=\text{red}}(\text{Parts})) \Join (\sigma_{\text{cost}<100}(\text{Catalog}) \Join \text{Suppliers}))) \cap$
   $(\pi_{\text{sname}}((\sigma_{\text{color}=\text{green}}(\text{Parts})) \Join (\sigma_{\text{cost}<100}(\text{Catalog}) \Join \text{Suppliers})))$
4. \( (\pi_{sid}(\sigma_{color='red'}(Parts)) \land (\sigma_{cost<100}(Catalog)) \land Suppliers)) \cup \\
(\pi_{sid}(\sigma_{color='green'}(Parts)) \land (\sigma_{cost<100}(Catalog)) \land Suppliers)) \\

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

**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?