EXERCISE 1 RA-TRC-SQL: step-by-step explanation

Consider the following schema:

SUPPLIERS(Sid, Name,City)
PARTS(Pid, Pname, Colour)
CATALOG(Sid, Pid, Cost)

where key attributes have been underlined; and relation CATALOG lists prices charged by Suppliers for Parts. Express in RA, TRC and SQL the queries that find the Names of suppliers who supply red parts.

POSSIBLE SOLUTIONS

1. Relational Algebra (RA):

CONSIDERATIONS: the information involved in this query is

- name of suppliers: stored in relation SUPPLIERS
- the color (red) of the supplied parts: stored in relation PARTS
- the information related to "who supplies what": stored in relation CATALOG

therefore, all three relations will have to participate in the query. The way we combine information stored in different relations in a query is by means of a Cartesian product or join operation.

In this case we use NATURAL JOIN (i.e., the implicit predicate is equality of all common attributes)

- PARTS ⋈ CATALOG ⋈ SUPPLIERS: this generates a big relation containing all the required information (and more)
- we are only interested in red part. Therefore we select on the basis of this condition: $\sigma_{Colour='red'}$ (PARTS⋈ CATALOG ⋈ SUPPLIERS))
- We only want to extract the Names of suppliers. Therefore we project on this attribute

π_{Name} ($\sigma_{Colour='red'}$ (PARTS⋈CATALOG ⋈ SUPPLIERS) **(SOLUTION 1)**

Possible optimised solution (tries to reduce the size of intermediate relations): for example, you can select red parts from relation PARTS before joining it with CATALOG.

π_{Name}(π_{Sid} (π_{Pid} ($\sigma_{Colour='red'}$ (PARTS))⋈CATALOG)⋈ SUPPLIERS) **(SOLUTION 2)**

2. Tuple Relational Calculus (TRC):

Analogous considerations can be made as before: all three relations will take part in the query. Therefore we need three variables corresponding to tuples of each relation. Also **in relational calculus we need to write the join predicate explicitely** (i.e., equality of common attributes). In this case the common attributes are:

- Sid common to SUPPLIERS and CATALOG
- Pid common to PARTS and CATALOG

{T:{Name} | (∃ T') (T'∈ SUPPLIERS ∧ (∃ X) (X∈ PARTS ∧ X[Colour]='red' ∧
(∃ Y) (Y∈ CATALOG ∧ Y[Pid]=X[Pid] ∧ Y[Sid]=T'[Sid])) ∧ T[Name]=T'[Name])}

where Y[Pid]=X[Pid] ∧ Y[Sid]=T'[Sid] is the **join predicate.**

3. SQL:

Also in SQL we need to write the join predicate explicitly

SELECT S.Name
FROM SUPPLIERS S, PARTS P, CATALOG C
WHERE **P.Colour='red' AND C.Pid=P.Pid AND C.Sid=S.Sid**