EXERCISES

Consider the following schema:

SUPPLIERS(<u>Sid</u>, Name,City)
PARTS(<u>Pid</u>, Pname, Colour)
CATALOG(<u>Sid, Pid</u>, Cost)

where key attributes have been underlined; and relation CATALOG lists prices charged by Suppliers for Parts. Express in RA and SQL the queries that find:

1. the Names of suppliers who supply red parts
2. the Sids of suppliers who supply red OR green parts
3. the Sids of suppliers who supply red parts AND live in Dublin
3*) the Sids of suppliers who supply red parts OR live in Dublin
4. the Sids of suppliers who supply red AND green parts
5. Pairs of Sids such that the supplier with the first Sid charges more than the supplier with the second Sid for the same part.
6. Pids of parts that are supplied by at least two different suppliers.

POSSIBLE SOLUTIONS
NOTE:
-   some solutions can be expressed also in alternative ways
-   the symbol ⋈ without any predicate specified indicates NATURAL JOIN (see definition)

1. RA:

    $\pi_{\text{Name}}(\pi_{\text{Sid}} (\pi_{\text{Pid}} (\sigma_{\text{Colour='red'}} (\text{PARTS})) \bowtie \text{CATALOG}) \bowtie \text{SUPPLIERS})$

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

2. RA:

    $\pi_{\text{Sid}} (\pi_{\text{Pid}} (\sigma_{\text{Colour='red'} \lor \text{Colour='green'}} (\text{PARTS})) \bowtie \text{CATALOG})$

    SQL:
    SELECT C.Sid
    FROM CATALOG C, PARTS P
    WHERE (P.Colour='red' OR P.Colour='green') AND C.Pid=P.Pid

3. RA:

    Let R and R' be:
    R=$\pi_{\text{Sid}} (\pi_{\text{Pid}} (\sigma_{\text{Colour='red'}} (\text{PARTS})) \bowtie \text{CATALOG})$; R'=$\pi_{\text{Sid}} (\sigma_{\text{City='Dublin'}} (\text{SUPPLIERS}))$

    The solution is: R $\cap$ R'

    SQL:
    SELECT S.Sid
    FROM SUPPLIERS S
    WHERE S.City='Dublin AND S.Sid **IN** (SELECT C.Sid
                                            FROM PARTS P, CATALOG C
                                            WHERE P.Colour='red' AND C.Pid=P.Pid)

***NOTE: "**IN**" is the same as "=ANY" in nested queries**

3*. RA:

Let R and R' be:
$R = \pi_{Sid} (\pi_{Pid} (\sigma_{Colour='red'} (PARTS)) \bowtie CATALOG); R' = \pi_{Sid} (\sigma_{City='Dublin'} (SUPPLIERS))$

The solution is: $R \cup R'$

SQL:
SELECT S.Sid
FROM SUPPLIERS S
WHERE S.City='Dublin OR S.Sid IN (SELECT C.Sid
                                     FROM PARTS P, CATALOG C
                                     WHERE P.Colour='red' AND C.Pid=P.Pid)

4. Similar

5. RA:

Let R' be:
$R' = \rho_{Sid' \leftarrow Sid, Pid' \leftarrow Pid, Cost' \leftarrow Cost} (CATALOG)$

The solution is: $\pi_{Sid,Sid'} (\sigma_{Pid=Pid' \wedge Sid <> Sid' \wedge Cost>Cost'} (CATALOG \ X \ R'))$

SQL:
SELECT C.Sid, C'.Sid
FROM CATALOG C, CATALOG C'
WHERE C.Pid=C'.Pid AND C.Sid <> C'.Sid AND C.Cost>C'.Cost

6. RA:

Let R' be:
$R' = \rho_{Sid' \leftarrow Sid, Pid' \leftarrow Pid} (CATALOG)$

The solution is: $\pi_{Pid} (\sigma_{Pid=Pid' \wedge Sid <> Sid'} (\pi_{Sid,Pid} (CATALOG) \ X \ R'))$

SQL:
SELECT distinct C.Pid
FROM CATALOG C, CATALOG C'
WHERE C.Pid=C'.Pid AND C.Sid <> C'.Sid