CS 631: DATA MANAGEMENT SYSTEMS DESIGN

ASSIGNMENT 2

EXERCISE 1 (SQL Queries)
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*)

The key fields are underlined, and the domain of each field is listed after the field name. Thus, SID is the key for SUPPLIERS, PID is the key for PARTS, and SID and PID together form the key for CATALOG. The CATALOG relation lists the prices charged for parts by suppliers. CATALOG.SID is a foreign key referring to SUPPLIERS.SID and CATALOG.PID is a foreign key referring to PARTS.PID.

Write the following queries in SQL.

1. Find the SIDs of suppliers who supply a red part and a green part.

```
SELECT C.SID
FROM CATALOG C, PART P
WHERE C.PID = P.PID AND P.COLOR = ‘red' AND
C.SID IN (SELECT C.SID
          FROM CATALOG C, PART P
          WHERE C.PID = P.PID AND P.COLOR = ‘green’)
```

```
(SELECT C.SID
FROM CATALOG C, PART P
WHERE C.PID = P.PID AND P.COLOR = ‘red')
INTERSECT
(SELECT C.SID
FROM CATALOG C, PART P
WHERE C.PID = P.PID AND P.COLOR = ‘green’)
```

```
(SELECT SID
FROM CATALOG NATURAL JOIN PART
WHERE COLOR = ‘red')
INTERSECT
(SELECT SID
FROM CATALOG NATURAL JOIN PART
WHERE COLOR = ‘green’)
```

2. Find the SIDs of suppliers who supply a red part or a green part.

```
SELECT C.SID
FROM CATALOG C, PART P
```

WHERE C.PID = P.PID **AND** (P.COLOR = ‘red’ **OR** P.COLOR = ‘green’

SELECT SID
FROM CATALOG **NATURAL JOIN** PART
WHERE COLOR = ‘red' **OR** P.COLOR = ‘green’

(**SELECT** C.SID
FROM CATALOG C, PART P
WHERE C.PID = P.PID **AND** P.COLOR = ‘red’)
UNION
(**SELECT** C.SID
FROM CATALOG C, PART P
WHERE C.PID = P.PID **AND** P.COLOR = ‘green’)

3. Find the SNAMEs of suppliers who supply every red part and every green part.

SELECT S.SID
FROM SUPPLIER S
WHERE NOT EXISTS ((**SELECT** P.PID
FROM PARTS P
WHERE P.COLOR=’red’ **OR** P.COLOR='green')
EXCEPT
(**SELECT** C.PID
FROM CATALOG C
WHERE C.SID = S.SID))

SELECT S.SID
FROM SUPPLIER S
WHERE NOT EXISTS (**SELECT** *
FROM PARTS P
WHERE P.COLOR = ‘red’ **OR** P.COLOR=’green’
AND NOT EXISTS (**SELECT** *
FROM CATALOG C
WHERE C.SID = S.SID **AND** C.PID = P.PID))

4. Find the SNAMEs of suppliers who do not supply every red part.

SELECT S.SNAME
FROM SUPPLIER S
WHERE EXISTS (**SELECT** *
FROM PARTS P
WHERE P.COLOR = ‘red’ **AND**
NOT EXISTS (**SELECT** *
FROM CATALOG C
WHERE C.SID = S.SID **AND** C.PID = P.PID))

Alternative: the following view computes the SIDs of suppliers who supply every red part.

```
CREATE VIEW SUPPLIERS_ALL_RED_PARTS
SELECT S.SID
FROM SUPPLIER S
WHERE NOT EXISTS ((SELECT P.PID
                   FROM PARTS P
                   WHERE P.COLOR='red')
                   EXCEPT
                  (SELECT C.PID
                   FROM CATALOG C
                   WHERE C.SID = S.SID))
```

Or:

```
CREATE VIEW SUPPLIERS_ALL_RED_PARTS
SELECT S.SID
FROM SUPPLIER S
WHERE NOT EXISTS (SELECT *
                  FROM PARTS P
                  WHERE P.COLOR = 'red' AND
                  NOT EXISTS (SELECT *
                              FROM CATALOG C
                              WHERE C.SID = S.SID AND C.PID = P.PID))
```

Then the answer to the query can be computed as follows.

```
SELECT SNAME
FROM SUPPLIER
WHERE SID NOT IN SUPPLIERS_ALL_RED_PARTS
```

5. For every supplier that only supplies red parts, print the name of the supplier and the average cost of parts that she supplies.

```
SELECT S.SNAME, AVG(COST) AS TOTAL
FROM SUPPLIER S, CATALOG C
WHERE S.SID = C.SID AND
C.SID NOT IN (SELECT C1.SID
              FROM CATALOG C1, PARTS P1
              WHERE C1.SID = S1.SID AND P1.COLOR <> 'Red')
GROUP BY S.SID, S.SNAME
```

```
SELECT SNAME, AVG(COST) AS TOTAL
FROM SUPPLIER NATURAL JOIN CATALOG
WHERE SID NOT IN (SELECT SID
                  FROM CATALOG NATURAL JOIN PARTS
                  WHERE COLOR <> 'Red')
GROUP BY SID, SNAME
```

6. For each part, find the SNAMEs of the suppliers who do not charge the most for that part. The answer of this query should have two columns: PID and SNAME.

```
SELECT C.PID, S.SNAME
FROM SUPPLIER S, CATALOG C
WHERE S.SID = C.SID AND
C.COST < (SELECT MAX(C1.COST)
          FROM CATALOG C1
          WHERE C1.PID = C.PID)
```

7. For every part supplied by a supplier who is at the city of Newark, print the PID and the SID and the name of the suppliers who sell it at the highest price.

```
SELECT C.PID, S.SID, S.SNAME
FROM SUPPLIERS S, CATALOG C
WHERE S.SID = C.SID AND
C.PID IN (SELECT PID
          FROM CATALOG C1, SUPPLIERS S1
          WHERE S1.SID = C1.SID AND
          S1.ADDRESS = 'Newark') AND
C.COST = (SELECT MAX(COST)
          FROM CATALOG
          WHERE C.PID = PID)
```

```
SELECT C.PID, S.SID, S.SNAME
FROM SUPPLIERS S, CATALOG C
WHERE S.SID = C.SID AND
          C.PID IN  (SELECT PID
                     FROM CATALOG NATURAL JOIN SUPPLIERS
                     WHERE ADDRESS = 'Newark') AND
          C.COST = (SELECT MAX(COST)
                     FROM CATALOG
                     WHERE PID = C.PID)
```

8. For every part which has at least two suppliers, find its PID, its PNAME and the total number of suppliers who sell it.

```
SELECT P.PID, P.PNAME, COUNT(C.SID)
FROM PARTS P, CATALOG C
WHERE P.PID = C.PID AND
GROUP BY P.PID
HAVING COUNT(C.SID) > = 2
```

```
SELECT PID, PNAME, COUNT(SID)
FROM PARTS NATURAL JOIN CATALOG
```

```
GROUP BY PID
HAVING COUNT(SID) > = 2
```

9. Find the PIDs of parts supplied by every supplier who is at the city of Newark or by every supplier who is at the city of Trenton.

```
(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS   (SELECT SID FROM SUPPLIERS WHERE ADDRESS = 'Newark'
                    EXCEPT
                   (SELECT    SID FROM CATALOG WHERE PID = P.PID)))
UNION
(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS   (SELECT SID FROM SUPPLIERS WHERE ADDRESS = 'Trenton'
                    EXCEPT
                   (SELECT    SID FROM CATALOG WHERE PID = P.PID)))

(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS (SELECT *
                  FROM SUPPLIERS S
                  WHERE S.ADDRESS = 'Newark' AND
                  NOT EXISTS (SELECT *
                              FROM CATALOG C
                              WHERE C.PID = P.PID AND C.SID = S.SID)))
UNION
(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS (SELECT *
                   FROM SUPPLIERS S
                  WHERE S.ADDRESS = 'Trenton' AND
                   NOT EXISTS (SELECT *
                              FROM CATALOG C
                              WHERE C.PID = P.PID AND C.SID = S.SID)))
```

10. Find the PIDs of parts supplied by every supplier who is at the city of Newark and by every supplier who is at the city of Trenton.

```
(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS   (SELECT SID
```

```
                    FROM SUPPLIERS
                    WHERE ADDRESS = 'Newark' OR ADDRESS = 'Trenton'
                    EXCEPT
                   (SELECT   SID
                    FROM     CATALOG
                    WHERE    PID = P.PID)))

(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS (SELECT *
                    FROM SUPPLIERS S
                    WHERE S.ADDRESS = 'Newark' OR S.ADDRESS = 'Trenton' AND
                    NOT EXISTS (SELECT *
                                FROM CATALOG C
                                WHERE C.PID = P.PID AND C.SID = S.SID)))

(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS   (SELECT SID FROM SUPPLIERS WHERE ADDRESS = 'Newark'
                    EXCEPT
                   (SELECT   SID FROM CATALOG WHERE PID = P.PID)))
INTERSECT
(SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS   (SELECT SID FROM SUPPLIERS WHERE ADDRESS = 'Trenton'
                    EXCEPT
                   (SELECT   SID FROM CATALOG WHERE PID = P.PID)))
```

11. Find the SIDs of suppliers who supply a red part but do not supply a blue part.

```
SELECT C.SID
FROM CATALOG C, PARTS P
WHERE C.PID = P.PID AND P.COLOR ='red' AND
C.SID NOT IN (SELECT C1.SID
              FROM CATALOG C1, PARTS P1
              WHERE C1.PID = P1.PID AND P1.COLOR ='blue')


(SELECT C.SID
FROM CATALOG C, PARTS P
WHERE C.PID = P.PID AND P.COLOR ='red')
EXCEPT
(SELECT C1.SID
FROM CATALOG C1, PARTS P1
WHERE C1.PID = P1.PID AND P1.COLOR ='blue')
```

```
SELECT SID
FROM PARTS NATURAL JOIN CATALOG
WHERE COLOR = ’red’ AND
        SID NOT IN (SELECT    SID
                    FROM      PARTS NATURAL JOIN CATALOG
                    WHERE     COLOR =’blue’)
```

```
(SELECT SID
FROM PARTS NATURAL JOIN CATALOG
WHERE COLOR = 'red')
EXCEPT
(SELECT SID
FROM PARTS NATURAL JOIN CATALOG
WHERE COLOR = 'blue')
```

12. For every supplier who supplies at least 4 parts, find his SID, SNAME and the PID of the most expensive part(s) that he supplies.

```
SELECT S.SID, S.SNAME, C.PID
FROM SUPPLIERS S, CATALOG C
WHERE S.SID=C.SID AND
S.SID IN (SELECT SID
          FROM CATALOG
          GROUP BY SID
          HAVING COUNT(PID) >= 4) AND
C.COST = (SELECT MAX(COST)
           FROM CATALOG
           WHERE SID = S.SID)
```

```
SELECT  S.SID, S.SNAME, C.PID
FROM    SUPPLIERS S, CATALOG C
WHERE   S.SID = C.SID AND
          (SELECT COUNT(PID)
          FROM CATALOG
          WHERE SID = S.SID) >= 4 AND
           C.COST = (SELECT MAX(COST)
                     FROM CATALOG
                     WHERE SID = S.SID)
```

13. For every distinct color of the parts, find the total number of suppliers who supply a part of this color.

```
SELECT P.COLOR, COUNT(C.SID)
FROM PARTS P, CATALOG C
WHERE P.PID = C.PID
GROUP BY P.COLOR

SELECT COLOR, COUNT(SID)
FROM PARTS NATURAL JOIN CATALOG
GROUP BY COLOR
```

14. Find the SIDs of suppliers who supply at least two parts of different color.

```
SELECT C.SID
FROM CATALOG C, PARTS P
WHERE C.PID=P.PID
GROUP BY C.SID
HAVING COUNT(DISTINCT P.COLOR) >= 2

SELECT SID
FROM PARTS NATURAL JOIN CATALOG
GROUP BY SID
HAVING COUNT(DISTINCT P.COLOR) >= 2

SELECT SID
FROM    (CATALOG NATURAL JOIN PARTS) NATURAL JOIN
        (CATALOG AS C1(PID, SID1, COST1) NATURAL JOIN PARTS AS  P1(PID1, PNAME1, COLOR1)))
WHERE COLOR <> COLOR1
```

15. For every part which has a supplier, find its PID, PNAME, its average cost, maximum cost and minimum cost.

```
SELECT P.PID, P.PNAME, MAX(C.COST), MIN(C.COST)
FROM PARTS P, CATALOG C
WHERE P.PID = C.PID
GROUP BY P.PID

SELECT      PID, PNAME, MAX(COST), MIN(COST)
FROM        PARTS NATURAL JOIN CATALOG
GROUP BY    PID
```