# WORKSHEET 7

**Student Name:** Vivek Kumar **UID:** 21BCS8129
**DOMAIN CAMP:** 16-01-2023 to 28-01-2023 **Section/Group:** DWWC-77
**Subject Name:** Database Management System

**Statement :-**

**An example database schema about movies is given below with underlined primary keys:**

- **Movie(<u>title: string, year: int, length</u>: int, genre: string, studioName: string, producerC#: int)**
- **MovieStar(<u>name: string</u>, address: string, gender: char, birthdate:date)**
- **StarsIn(<u>movieTitle: string, movieYear: string, starName: string</u>)**
- **MovieExec(name: string, address: string, <u>CERT#: int</u>, netWorth:int) • Studio(<u>name: string</u>, address: string, presC#: int)**

**Ques 1:- Display all those names of stars and addresses which are not having net worth recorded in the database.**

$$(\pi_{name,address}(MovieStar) - \pi_{name,address}(MovieExec))$$

**Ques 2:- Consider the following SELECT-FROM-WHERE statement.**

**SELECT movieTitle**

**FROM StarsIn, MovieStar**

**WHERE starName = name AND birthdate = 1960 Write**

**a similar query in the form of relational algebra:**

$$\pi_{movieTitle}\ \sigma_{StarName=name \wedge birthdate=1960}(StarsIn \times MovieStar)$$

**Ques3:- Write a query using movie database to display the names and total length of movies where producer has produced a movie before 1930.**

$$\pi_{name,SUM(length)}\sigma_{MIN(year)<1930}\gamma_{name,MIN(year),SUM(length)}$$

$$\sigma_{cert\#=producerC\#}(MovieExec \times Movie)$$

**Statement :-**

**Consider the following schema:**

**Suppliers(<u>sid: integer</u>, sname: string, address: string)**

**Parts(<u>pid: integer</u>, pname: string , color: string)**

**Catalog(<u>sid: integer</u>, pid: integer, cost: real)**

The key fields are underlined, and the domain of each field is listed after the field name. Therefore 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. Write the following queries in relational algebra& corresponding SQL statement:

**Ques 4. Find the names of suppliers who supply some red part.**

Relation Algebra Query (RA):

$$\pi_{Sname}(\pi_{Sid}((\pi_{pid}\sigma_{color='red'}\ Parts) \bowtie Catalog) \bowtie Suppliers)$$

**SQL Query:**
SELECT S.sname
FROM Suppliers S, Parts P, Catalog C
WHERE P.color='red' AND C.pid=P.pidAND C.sid=S.sid

**Ques 5:- Find the sids of suppliers who supply some red or green part.** Relational

Algebra

$$\pi_{sid}(\pi_{pid}(\sigma_{color='red' \vee color='green'}\ Parts) \bowtie catalog)$$

**SQL Query:**
SELECT C.sid
FROM Catalog C, Parts P
WHERE (P.color = 'red' OR P.color = 'green')

AND P.pid = C.pid

**Ques 6:-Find the sids of suppliers who supply some red part or are at 221 Packer Street.** Relational

Algebra

$$\rho(R1, \pi_{sid}((\pi_{pid}\sigma_{color='red'}Parts) \bowtie Catalog))$$
$$\rho(R2, \pi_{sid}\sigma_{address='221PackerStreet'}Suppliers)$$
$$R1 \cup R2$$

**SQL Query**

```
SELECT S.sid
FROM Suppliers S
WHERE S.address = '221 Packer street'
OR S.sidIN ( SELECTC.sid
FROM Parts P, Catalog C
WHERE P.color='red' AND P.pid = C.pid)
```

**Ques 7:-Find the sids of suppliers who supply some red part and some green part.** Relational

Algebra

$$\rho(R1, \pi_{sid}((\pi_{pid}\sigma_{color='red'} Parts) \bowtie Catalog))$$
$$\rho(R2, \pi_{sid}((\pi_{pid}\sigma_{color='green'} Parts)\bowtie Catalog))$$
$$R1 \cap R2$$

**SQL Query**

```
SELECT C.sid
FROM Parts P, Catalog C
WHERE P.color = 'red' AND P.pid = C.pid
AND EXISTS ( SELECTP2.pid
FROM Parts P2, Catalog C2
WHERE P2.color = 'green' AND C2.sid = C.sid
AND P2.pid = C2.pid)
```

**Ques 9:-Find the sids of suppliers who supply every part.**

**Relational Algebra**

$$(\pi_{sid,pid}Catalog)/(\pi_{pid}Parts)$$

**SQL Query:**

```
SELECT C.sid
FROM Catalog C
WHERE NOT EXISTS (SELECT P.pid
FROM Parts P
WHERE NOT EXISTS (SELECT C1.sid
FROM Catalog C1
WHERE C1.sid = C.sid
AND C1.pid = P.pid))
```

**Ques 10:-Find the sids of suppliers who supply every red part.**

Relational Algebra

$$(\pi_{sid,pid}Catalog)/(\pi_{pid}\sigma_{color='red'}Parts)$$

**SQL Query**

```
SELECT C.sid
FROM Catalog C
WHERE NOT EXISTS (SELECT P.pid
FROM Parts P
WHERE P.color = ‘red’
AND (NOT EXISTS (SELECT C1.sid
FROM Catalog C1
WHERE C1.sid = C.sidAND C1.pid
= P.pid)))
```

**Ques 11:-Find the sids of suppliers who supply every red or green part.**

Relational Algebra

$$(\pi_{sid,pid}Catalog)/(\pi_{pid}\sigma_{color='red' \lor color='green'}Parts)$$

**SQL Query**

SELECT C.sid
FROM Catalog C
WHERE NOT EXISTS (SELECT P.pid FROM
Parts P
WHERE (P.color = 'red' OR P.color = 'green')
AND (NOT EXISTS (SELECT C1.sid
FROM Catalog C1
WHERE C1.sid = C.sidAND
C1.pid = P.pid)))

**Ques 12:-Find the sids of suppliers who supply every red part or supply every green part.** Relational

Algebra

$$\rho(R1, ((\pi_{sid,pid}Catalog)/(\pi_{pid}\sigma_{color='red'}Parts))) \ \rho(R2, ((\pi_{sid,pid}Catalog)/(\pi_{pid}\sigma_{color='green'}Parts)))$$
$$R1 \cup R2$$

**SQL Query**

SQL
SELECT C.sid
FROM Catalog C
WHERE (NOT EXISTS (SELECT P.pid
FROM Parts P
WHERE P.color = 'red' AND
(NOT EXISTS (SELECT C1.sid
FROM Catalog C1
WHERE C1.sid = C.sid AND
C1.pid = P.pid))))
OR ( NOT EXISTS (SELECT P1.pid FROM
Parts P1
WHERE P1.color = 'green' AND
(NOT EXISTS (SELECT C2.sid
FROM Catalog C2
WHERE C2.sid = C.sid AND
C2.pid = P1.pid))))

**Ques 13:-Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.** Relational Algebra

$$\rho(R1,Catalog)\ \rho(R2,Catalog)$$
$$\pi_{R1.sid,R2.sid}(\sigma_{R1.pid=R2.pid \wedge R1.sid\_=R2.sid \wedge R1.cost>R2.cost}(R1 \times R2))$$

**SQL Query**

SELECT C1.sid, C2.sid
FROM Catalog C1, Catalog C2
WHERE C1.pid = C2.pid AND C1.sid = C2.sid
AND C1.cost >C2.cost

**Ques 14:-Find the pids of parts supplied by at least two different suppliers.** Relational

Algebra

$$\rho(R1,Catalog)\ \rho(R2,Catalog)$$
$$\pi_{R1.pid}\sigma_{R1.pid=R2.pid \wedge R1.sid\_=R2.sid}(R1 \times R2)$$

**SQL Query**

SELECT C.pid
FROM Catalog C
WHERE EXISTS (SELECT C1.sid
FROM Catalog C1
WHERE C1.pid = C.pidAND C1.sid = C.sid)

**Ques 15:- Find the pids of the most expensive parts supplied by suppliers named Yosemite Sham.**

Relational Algebra

$$\rho(R1,\ \pi_{sid}\sigma_{sname='YosemiteSham'}Suppliers)$$
$$\rho(R2,R1 \bowtie Catalog)$$
$$\rho(R3,R2)$$
$$\rho(R4(1 \to sid,\ 2 \to pid,\ 3 \to cost),\ \sigma_{R3.cost<R2.cost}(R3 \times R2))\ \pi_{pid}(R2$$
$$-\ \pi_{sid,pid,cost}R4)$$

**SQL Query**

SQL
SELECT C.pid
FROM Catalog C, Suppliers S

WHERE S.sname = ‘Yosemite Sham’ AND C.sid = S.sid
AND C.cost≥ ALL (Select C2.cost
FROM Catalog C2, Suppliers S2 WHERE
S2.sname = ‘Yosemite Sham’
AND C2.sid = S2.sid)

**Ques16:- Consider the Supplier-Parts-Catalog schema from the previous question. State what the following queries compute:**

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

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

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

   $(\pi_{sname}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$

4. $(\pi_{sid}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)) \cap$

   $(\pi_{sid}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$

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

   $(\pi_{sid,sname}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)))$

The statements can be interpreted as:

1. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars.

2. This Relational Algebra statement does not return anything because of the sequence of projection operators. Once the sid is projected, it is the only field in the set. Therefore, projecting on sname will not return anything.

3. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

4. Find the Supplier ids of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

5. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

**Q17. Consider the following schemas:**

**Suppliers(<u>sid: integer</u>, sname: string, address: string)**
**Parts(<u>pid: integer</u>, pname: string, color: string)**
**Catalog(<u>sid: integer</u>, pid: integer, cost: real)**

**State what the following queries compute:**

1. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog)) \bowtie Suppliers)$
2. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$
3. $(\pi_{sname}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)) \cap$
$(\pi_{sname}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$
4. $(\pi_{sid}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)) \cap$
$(\pi_{sid}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$
5. $\pi_{sname}((\pi_{sid,sname}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)) \cap$
$(\pi_{sid,sname}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)))$

1. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars.

2. This Relational Algebra statement does not return anything because of the sequence of projection operators. Once the sid is projected, it is the only field in the set. Therefore, projecting on sname will not return anything.

3. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

4. Find the Supplier ids of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

5. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

**Statement:-**

**Consider the following relations containing airline flight information:**

**Flights (<u>flno: integer</u>, from: string, to: string, distance: integer, departs: time, arrives: time)**
**Aircraft (<u>aid: integer</u>, aname: string, cruisingrange: integer)**
**Certified (<u>eid: integer, aid: integer</u>)**
**Employees (<u>eid: integer</u>, ename: string, salary: integer)**
**The key fields are underlined. Write the following queries in tuple relational calculus, and domain relational calculus:**

**Q18. Find the eids of pilots certified for some Boeing aircraft.**

TRC

{C.eid | C ∈ Certif ied ∧ ∃A ∈ Aircraf t(A.aid = C.aid ∧ A.aname = ‘Boeing )}

DRC

{Ceid|Ceid, Caid ∈ Certif ied ∧ ∃Aid, AN, AR(Aid, AN, AR ∈ Aircraf t ∧Aid = Caid ∧ AN = ‘Boeing )}

SQL

SELECT C.eid FROM Aircraft A, Certified C WHERE A.aid = C.aid AND A.aname = ‘Boeing

**Q19. Find the names of pilots certified for some Boeing aircraft.**

TRC

{E.ename | E ∈ Employees ∧ ∃C ∈ Certif ied (∃A ∈ Aircraf t(A.aid = C.aid ∧ A.aname = ‘Boeing ∧ E.eid = C.eid))}

DRC

{EN|Eid, EN, ES ∈ Employees∧ ∃Ceid, Caid(Ceid, Caid ∈ Certif ied∧ ∃Aid, AN, AR(Aid, AN, AR ∈ Aircraf t∧ Aid = Caid ∧ AN = ‘Boeing ∧ Eid = Ceid)}

SQL

SELECT E.ename FROM Aircraft A, Certified C, Employees E WHERE A.aid = C.aid AND A.aname = ‘Boeing’ AND E.eid = C.eid

**Q20. Find the aids of all aircraft that can be used on non-stop flights from Bonn to Madras.**

TRC

{A.aid | A ∈ Aircraf t ∧ ∃F ∈ Flights (F.from = ‘Bonn ∧ F.to = ‘M adrid ∧ A.cruisingrange > F.distance)}

DRC

{Aid | Aid, AN, AR ∈ Aircraf t∧ (∃F N, F F, F T, FDi, FDe, F A(F N, F F, F T, FDi, FDe, F A ∈ Flights∧ F F = ‘Bonn ∧ F T = ‘M adrid ∧ F Di < AR))}

SQL

SELECT A.aid FROM Aircraft A, Flights F WHERE F.from = ‘Bonn’ AND F.to = ‘Madrid’ AND A.cruisingrange > F.distance

**Q21. Identify the flights that can be piloted by every pilot whose salary is more than $100,000. Solution:**

TRC

{F.flno | F ∈ Flights ∧ ∃A ∈ Aircraf t∃C ∈ Certif ied ∃E ∈ Employees(A.cruisingrange > F.distance ∧ E.salary > 100, 000∧ A.aid = C.aid ∧ E.eid = C.eid)}

DRC

{F N | F N, F F, F T, FDi, FDe, F A ∈ Flights∧ ∃Ceid, Caid(Ceid, Caid ∈ Certif ied∧ ∃Aid, AN, AR(Aid, AN, AR ∈ Aircraf t∧ ∃Eid, EN, ES(Eid, EN, ES ∈ Employees (AR > F Di ∧ ES > 100, 000
∧ Aid = Caid ∧ Eid = Ceid)}

SQL

SELECT E.ename FROM Aircraft A, Certified C, Employees E, Flights F WHERE A.aid = C.aid AND E.eid = C.eid AND distance < cruisingrange AND salary > 100,000

**Q22. Find the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.**

TRC

{E.ename | E ∈ Employees ∧ ∃C ∈ Certif ied(∃A ∈ Aircraf t (A.aid = C.aid ∧ E.eid = C.eid ∧

A.cruisingrange > 3000))∧ ¬(∃C2 ∈ Certif ied(∃A2 ∈ Aircraf t(A2.aname = 'Boeing ∧ C2.aid = A2.aid ∧ C2.eid = E.eid)))}

DRC

{EN|Eid, EN, ES ∈ Employees∧ ∃Ceid, Caid(Ceid, Caid ∈ Certif ied∧ ∃Aid, AN, AR(Aid, AN, AR ∈ Aircraf t∧ Aid = Caid ∧ Eid = Ceid ∧ AR > 3000))∧ ¬(∃Aid2, AN2, AR2(Aid2, AN2, AR2 ∈ Aircraf t∧ ∃Ceid2, Caid2(Ceid2, Caid2 ∈ Certif ied ∧Aid2 = Caid2 ∧ Eid = Ceid2 ∧ AN2='Boeing )))}

SQL

SELECT E.ename FROM Certified C, Employees E, Aircraft A WHERE A.aid = C.aid AND E.eid = C.eid AND A.cruisingrange > 3000 AND E.eid NOT IN ( SELECT C2.eid FROM Certified C2, Aircraft A2 WHERE C2.aid = A2.aid AND A2.aname = 'Boeing' )

**Statement:-**

**Consider the following relations containing airline flight information:**

**Flights(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price number)**
**Aircraft(aid: integer, aname: string, cruisingrange: integer)**
**Certified(eid: integer, aid: integer)**
**Employees(eid: integer, ename: string, salary: integer)**

**The key fields are underlined. Write the following queries in tuple relational calculus, and domain relational calculus:**

**Q23. Find the eids of employees who make the highest salary.**

TRC

{E1.eid | E1 ∈ Employees∧¬(∃E2 ∈ Employees(E2.salary > E1.salary))}

DRC

{<Eid1>|<Eid1,EN1,ES1>∈ Employees∧ ¬(∃Eid2,EN2,ES2(<Eid2,EN2,ES2> ∈ Employees ∧ ES2 > ES1))}

SQL

SELECT E.eid FROM Employees E WHERE E.salary = ( Select MAX (E2.salary) FROM Employees E2 )

**Q24. Find the eids of employees who make the second highest salary.**

TRC

$$\{E1.eid \mid E1 \in Employees \wedge \exists E2 \in Employees(E2.salary > E1.salary \wedge \neg(\exists E3 \in Employees(E3.salary > E2.salary)))\}$$

DRC

$$\{\langle Eid1\rangle \mid \langle Eid1,EN1,ES1\rangle \in Employees \wedge \exists Eid2,EN2,ES2(\langle Eid2,EN2,ES2\rangle \in Employees(ES2 > ES1) \wedge \neg(\exists Eid3,EN3,ES3(\langle Eid3,EN3,ES3\rangle \in Employees(ES3 > ES2))))\}$$

SQL

```
SELECT E.eid FROM Employees E WHERE E.salary = (SELECT MAX (E2.salary) FROM Employees E2 WHERE E2.salary = (SELECT MAX (E3.salary) FROM Employees E3 ))
```

**Q25. Find the eids of employees who are certified for the largest number of aircraft.**

This cannot be expressed in relational calculus because there is no operator to count, and this query requires the ability to count up to a number that depends on the data. The query can however be expressed in SQL as follows:

```
SELECT Temp.eid FROM ( SELECT C.eid AS eid, COUNT (C.aid) AS cnt, FROM Certified C GROUP BY C.eid) AS Temp WHERE Temp.cnt = ( SELECT MAX (Temp.cnt) FROM Temp)
```

**Q26. Find the eids of employees who are certified for exactly three aircraft.**

TRC

$$\{C1.eid \mid C1 \in Certif\ ied \wedge \exists C2 \in Certif\ ied(\exists C3 \in Certif\ ied\ (C1.eid = C2.eid \wedge C2.eid = C3.eid \wedge C1.aid = C2.aid \wedge C2.aid = C3.aid \wedge C3.aid = C1.aid \wedge \neg(\exists C4 \in Certif\ ied\ (C3.eid = C4.eid \wedge C1.aid = C4.aid \wedge C2.aid = C4.aid \wedge C3.aid = C4.aid))))\}$$

DRC

$$\{\langle CE1\rangle \mid \langle CE1,CA1\rangle \in Certif\ ied \wedge \exists CE2,CA2(\langle CE2,CA2\rangle \in Certif\ ied \wedge \exists CE3,CA3(\langle CE3,CA3\rangle \in Certif\ ied \wedge (CE1 = CE2 \wedge CE2 = CE3 \wedge CA1 = CA2 \wedge CA2 = CA3 \wedge CA3 = CA1 \wedge \neg(\exists CE4,CA4(\langle CE4,CA4\rangle \in Certif\ ied \wedge (CE3 = CE4 \wedge CA1 = CA4 \wedge CA2 = CA4 \wedge CA3 = CA4))))\}$$

SQL

SELECT C1.eid FROM Certified C1, Certified C2, Certified C3 WHERE (C1.eid = C2.eid AND C2.eid = C3.eid AND C1.aid = C2.aid AND C2.aid = C3.aid AND C3.aid = C1.aid) EXCEPT SELECT C4.eid FROM Certified C4, Certified C5, Certified C6, Certified C7, WHERE (C4.eid = C5.eid AND C5.eid = C6.eid AND C6.eid = C7.eid AND C4.aid = C5.aid AND C4.aid = C6.aid AND C4.aid = C7.aid AND C5.aid = C6.aid AND C5.aid = C7.aid AND C6.aid = C7.aid ) This could also be done in SQL using COUNT.

**Q27. Find the total amount paid to employees as salaries.**

TRC

{C1.eid | C1 ∈ Certif ied ∧ ∃C2 ∈ Certif ied(∃C3 ∈ Certif ied (C1.eid = C2.eid ∧ C2.eid = C3.eid∧ C1.aid = C2.aid ∧ C2.aid = C3.aid ∧ C3.aid = C1.aid∧ ¬(∃C4 ∈ Certif ied (C3.eid = C4.eid ∧ C1.aid = C4.aid∧ C2.aid = C4.aid ∧ C3.aid = C4.aid))))}

DRC

{<CE1>|<CE1,CA1> ∈ Certif ied∧ ∃CE2,CA2(<CE2,CA2> ∈ Certif ied∧ ∃CE3,CA3(<CE3,CA3> ∈ Certif ied∧ (CE1 = CE2 ∧ CE2 = CE3∧ CA1 = CA2 ∧ CA2 = CA3 ∧ CA3 = CA1∧ ¬(∃CE4,CA4(<CE4,CA4> ∈ Certif ied∧ (CE3 = CE4 ∧ CA1 = CA4∧ CA2 = CA4 ∧ CA3 = CA4))))}

SQL

SELECT C1.eid FROM Certified C1, Certified C2, Certified C3 WHERE (C1.eid = C2.eid AND C2.eid = C3.eid AND C1.aid = C2.aid AND C2.aid = C3.aid AND C3.aid = C1.aid) EXCEPT SELECT C4.eid FROM Certified C4, Certified C5, Certified C6, Certified C7, WHERE (C4.eid = C5.eid AND C5.eid = C6.eid AND C6.eid = C7.eid AND C4.aid = C5.aid AND C4.aid = C6.aid AND C4.aid = C7.aid AND C5.aid = C6.aid AND C5.aid = C7.aid AND C6.aid = C7.aid ) This could also be done in SQL using COUNT.

1. This cannot be expressed in relational algebra (or calculus) because there is no operator to sum values. The query can however be expressed in SQL as follows:

   SELECT SUM (E.salaries) FROM Employees E

**Q28. Let R = (A, B, C), and let r1 and r2 both be relations on schema R. Give an expression in the domain relational calculus that is equivalent to each of the following: a. $\Pi A(r1)$**

**b. $\sigma B = 17\ (r1)$**

**c. $r1 \cup r2$**

**d.** $r1 \cap r2$

**e.** r1 − r2
**f. ΠA,B(r1) ΠB,C (r2)**

a. {< t > | ∃ p, q (< t, p, q > ∈ r1)}
b. {< a, b, c > | < a, b, c > ∈ r1 ∧ b = 17}
c. {< a, b, c > | < a, b, c > ∈ r1 ∨ < a, b, c > ∈ r2}
d. {< a, b, c > | < a, b, c > ∈ r1 ∧ < a, b, c > ∈ r2}
e. {< a, b, c > | < a, b, c > ∈ r1 ∧ < a, b, c > ∈ r2}
f. {< a, b, c > | ∃ p, q (< a, b, p > ∈ r1 ∧ < q, b, c > ∈ r2)

**Statement:-**
**Consider the following schemas:**

**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. Write the following queries in tuple relational calculus, domain relational calculus and SQL Statement:**

**Q29. Find the names of suppliers who supply some red part.**

TRC

{T | ∃T 1 ∈ Suppliers(∃X ∈ P arts(X.color = red ∧ ∃Y ∈ Catalog (Y.pid = X.pid ∧ Y.sid = T 1.sid)) ∧ T.sname = T 1.sname)}

DRC

{Y |X, Y, Z ∈ Suppliers ∧ ∃P, Q, R(P, Q, R ∈ P arts ∧R = red ∧ ∃I, J, K(I, J, K ∈ Catalog ∧ J = P ∧ I = X))}

SQL

SELECT S.sname FROM Suppliers S, Parts P, Catalog C WHERE P.color='red' AND C.pid=P.pid AND C.sid=S.sid

**Q30. Find the sids of suppliers who supply some red part and some green part.**

TRC

{T | ∃T 1 ∈ Catalog(∃X ∈ P arts(X.color = ‘red ∧ X.pid = T 1.pid) ∧∃T 2 ∈ Catalog(∃Y ∈ P arts(Y.color = green ∧ Y.pid = T 2.pid) ∧T 2.sid = T 1.sid) ∧ T.sid = T 1.sid)}

DRC

{X|X, Y, Z ∈ Catalog ∧ ∃A, B, C(A, B, C ∈ P arts ∧C = red ∧ A = Y ) ∧∃P, Q, R(P, Q, R ∈ Catalog ∧ ∃E, F, G(E, F, G ∈ P arts ∧G = green ∧ E = Q) ∧ P = X)}

SQL

SELECT C.sid FROM Parts P, Catalog C WHERE P.color = ‘red’ AND P.pid = C.pid AND EXISTS ( SELECT P2.pid FROM Parts P2, Catalog C2 WHERE P2.color = ‘green’ AND C2.sid = C.sid AND P2.pid = C2.pid )