

### \* Lab Exercise-4:

- Aim: To create the following tables in the sailors-Boats database

Sailors(sid:integer, sname:string, rating:integer, age:real)

Boats(bid:integer, bname:string, color:string)

Reserves(sid:integer, bid:integer, day:date)

and perform the given queries.

- SQL Commands:

#### → Creation of Database

mysql> CREATE DATABASE Sailors-Boats;

mysql> USE Sailors-Boats;

#### → Creation of tables

mysql> CREATE TABLE Sailors( > sid INT PRIMARY KEY, > sname VARCHAR(15), > rating INT, > age REAL > );

mysql> CREATE TABLE Boats( > bid INT PRIMARY KEY, > bname VARCHAR(15), > color VARCHAR(15) > );

```
mysql> CREATE TABLE Reserves (
    >     sid INT PRIMARY KEY,
    >     bid INT PRIMARY KEY,
    >     day DATE,
    >     FOREIGN KEY(sid) REFERENCES
    >                           Sailors(sid),
    >     FOREIGN KEY(bid) REFERENCES
    >                           Boats(bid)
    > );
```

→ Inserting the values into Sailors

```
mysql> INSERT INTO Sailors VALUES
    > (22,"Dustin",7,45), (29,"Brutus",1,33),
    > (31,"Lubber",8,55.5), (32,"Andy",8,25.5),
    > (58,"Rusty",10,35), (64,"Horatio",7,35),
    > (71,"Zorba",10,16), (74,"Horatio",9,35),
    > (85,"Art",3,25.5), (95,"Bob",3,63.5);
```

```
mysql> SELECT * FROM Sailors;
```

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

→ Inserting the values into Boats

mysql> INSERT INTO Boats VALUES

> (101, "Interlake", "blue"), (102, "Interlake", "red"),  
 > (103, "clipper", "green"), (104, "Marine", "red");

mysql> SELECT \* FROM Boats;

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	clipper	green
104	Marine	red

→ Inserting the values into Reserves

mysql> INSERT INTO Reserves VALUES

> (22, 101, '1998-10-10'), (22, 102, '1998-10-10'),  
 > (22, 103, '1998-08-10'), (22, 104, '1998-07-10'),  
 > (31, 102, '1998-10-11'), (31, 103, '1998-06-11'),  
 > (31, 104, '1998-12-11'), (64, 101, '1998-05-09'),  
 > (64, 102, '1998-08-09'), (74, 103, '1998-08-09');

mysql> SELECT \* FROM Reserves;

sid	bid	day
22	101	1998-10-10
22	102	1998-10-10
22	103	1998-08-10
22	104	1998-07-10
31	102	1998-10-11
31	103	1998-06-11
31	104	1998-12-11
64	101	1998-05-09
64	102	1998-08-09
74	103	1998-08-09

Q1) Find the names of sailor who have reserved boat 103.

```
mysql > SELECT s.sname
      → FROM Sailors S, Reserves R
      → WHERE S.sid = R.sid
      → AND R.bid = 103;
```

sname
Dustin
Lubber
Horatio

Q2) Find the names of sailors who have reserved a red boat.

```
mysql > SELECT DISTINCT s.sname
      → FROM Sailors S, Reserves R, Boats b
      → WHERE S.sid = R.sid AND
      → AND R.bid = b.bid AND b.color = "red";
```

sname
Dustin
Lubber
Horatio

Q3) Find the colors of boats reserved by Lubber.

```
mysql > SELECT DISTINCT b.color
      → FROM Sailors S, Reserves R, Boats b
      → WHERE S.sid = R.sid AND
      → R.bid = b.bid AND S.sname = "Lubber";
```

Color
Red
green

Q4) Find the names of sailors who have reserved atleast one boat.

```
mysql > SELECT DISTINCT S.sname
      → FROM Sailors S, Reserves R
      → WHERE S.sid = R.sid;
```

sname
Dustin
Lubber
Horatio

Q5) Find the names of sailor who have reserved a red (or) green boat.

```
mysql > SELECT DISTINCT S.sname
      → FROM Sailors S, Reserves R, Boats b
      → WHERE S.sid = R.sid AND
      → R.bid = b.bid AND b.color IN("red", "green");
```

sname
Dustin
Lubber
Horatio

Q6) Find the names of sailors who have reserved atleast 2 boats.

```
mysql > SELECT DISTINCT S.sname
      → FROM Sailors S, Reserves R
      → WHERE S.sid = R.sid
      → GROUP BY S.sname
      → HAVING COUNT(R.bid) ≥ 2;
```

sname
Dustin
Horatio
Lubber

Q7) Find the names of sailors who have reserved a red & a green boat.

```
mysql> SELECT DISTINCT S.sname
    → FROM Sailors S
    → JOIN Reserves R ON S.sid=R.sid
    → JOIN Boats B ON R.bid=B.bid
    → WHERE B.color="red"
    → AND S.sid IN (
        →     SELECT R2.sid
        →     FROM Reserves R2
        →     JOIN Boats b ON R2.bid=b.bid
        →     WHERE b.color="green"
    → );
```

sname
Dustin
Lubber

Q8) Find the sids of sailors with age over 20 who have not reserved a red boat.

```
mysql> SELECT S.sid
    → FROM Sailors S
    → WHERE S.age > 20
    → AND S.sid NOT IN (
        →     SELECT R.sid
        →     FROM Reserves R,Boats B
        →     WHERE R.bid=B.bid
        →     AND B.color="red"
    → );
```

sid
29
32
58
74
85
95

Q9) Find the names of Sailors who have reserved all boats.

```
mysql > SELECT S.sname
      → FROM Sailors S, Reserves R
      → WHERE S.sid = R.sid
      → GROUP BY S.sid, S.sname
      → HAVING COUNT(DISTINCT R.bid) = 4;
```

sname
Dustin

Q10) Find the names of Sailors who have reserved all boats called INTERLAKE.

```
mysql > SELECT S.sname
      → FROM Sailors S, Reserves R, Boats b
      → WHERE S.sid = R.sid AND R.bid = b.bid
      → AND b.bname = "Interlake"
      → GROUP BY S.sname
      → HAVING COUNT(b.bid) = (
      →     SELECT COUNT(DISTINCT b2.bid)
      →     FROM Boats b2
      →     WHERE b2.name = "Interlake"
      → );
```

sname
Dustin
Horatio

### Result :

Successfully created the Sailors, Boats, Reserves tables in Sailors\_Boats database and performed the given queries.

### \* Lab Exercise - 5:

Aim: To write the relational Algebra queries for the given Questions.

Q1) Find the names of sailors who have reserved a boat 103.

$$\Pi_{\text{sname}}(\sigma_{\text{bid}=103}(\text{Sailors} \bowtie \text{Reserves}))$$

Q2) Find the names of sailors who have reserved a red boat.

$$\Pi_{\text{sname}}(\sigma_{\text{color}=\text{'red'}}(\text{Boats} \bowtie \text{Reserves} \bowtie \text{Sailors}))$$

Q3) Find the colors of Boats reserved by Lubber

$$\Pi_{\text{color}}(\sigma_{\text{sname}=\text{"Lubber"}}(\text{Boats} \bowtie \text{Reserves} \bowtie \text{Sailors}))$$

Q4) Find the names of sailors who have reserved atleast one boat.

$$\Pi_{\text{sname}}(\text{Sailors} \bowtie \text{Reserves})$$

Q5) Find the names of sailors who have reserved a red (or) a green boat.

$$\Pi_{\text{sname}}(\sigma_{\text{color}=\text{'red'} \vee \text{color}=\text{'green'}}(\text{Boats} \bowtie \text{Reserves} \bowtie \text{Sailors}))$$

Q6) Find the names of sailors who have reserved atleast two boats.

$$\Pi_{\text{sname}}(\sigma_{\text{Count}(\text{bid}) \geq 2}(\text{Reserves} \bowtie \text{Sailors}) \text{ Group By sid })$$

Q7) Find the names of sailors who have reserved a red and green boat.

$$\Pi_{\text{sname}}(\sigma_{\text{color}=\text{'red'}}(\text{Boats}) \bowtie \text{Reserves} \bowtie \text{Sailors})$$

∩

$$\Pi_{\text{sname}}(\sigma_{\text{color}=\text{'green'}}(\text{Boats}) \bowtie \text{Reserves} \bowtie \text{Sailors})$$

Q8) Find the sids of sailors with age over 20 who have not reserved a red boat.

$$\Pi_{\text{sid}}(\sigma_{\text{age} > 20}(\text{Sailors})) - \Pi_{\text{sid}}(\sigma_{\text{color}=\text{'red'}}(\text{Boats}) \bowtie \text{Reserves} \bowtie \text{Sailors})$$

Q9) Find the names of sailors who have reserved all boats.

$$e(\text{TempSids}, (\Pi_{\text{sid,bid}} \text{Reserves}) / (\Pi_{\text{bid}} \text{Boats}))$$

$$\Pi_{\text{sname}}(\text{TempSids} \bowtie \text{Sailors})$$

Q10) Find the names of sailors who have reserved all boats called Interlake.

$$e(\text{TempSids}, (\Pi_{\text{sid,bid}} \text{Reserves}) / (\Pi_{\text{bid}}(\sigma_{\text{bname}=\text{'Interlake'}}(\text{Boats}))))$$

$$\Pi_{\text{sname}}(\text{TempSids} \bowtie \text{Sailors})$$

#### Result :

Successfully written the Relational Algebra queries for the given Questions.

\* Lab Exercise - 6:

Aim: To write a Tuple Relational Calculus for the given queries.

Q1) Find the Sailors with a rating above 7.

$$\{ s \mid s \in \text{Sailors} \wedge s.\text{rating} > 7 \}$$

Q2) Find the names and ages of sailors with a rating above 7.

$$\{ P \mid \exists s \in \text{Sailors} (s.\text{rating} > 7 \wedge P.sname = s.sname \wedge P.age = s.age) \}$$

Q3) Find the Sailor name, boat id and reservation date for each reservation

$$\{ P \mid \exists R \in \text{Reserves} \exists s \in \text{Sailors} (R.sid = s.sid \wedge P.bid = R.bid \wedge P.day = R.day \wedge P.sname = s.sname) \}$$

Q4) Find the names of sailors who have reserved boat 103.

$$\{ P \mid \exists s \in \text{Sailors} \exists R \in \text{Reserves} (R.sid = s.sid \wedge R.bid = 103 \wedge P.sname = s.sname) \}$$

Q5) Find the names of Sailors who have reserved a red boat.

$$\{ P \mid \exists s \in \text{Sailors} \exists R \in \text{Reserves} (R.sid = s.sid \wedge P.sname = s.sname \wedge \exists B \in \text{Boats} (B.bid = R.bid \wedge B.color = "red")) \}$$

Q6) Find the names of sailors who have reserved atleast two boats.

{ P |  $\exists s \in \text{sailors } \exists r_1 \in \text{Reserves } \exists r_2 \in \text{Reserves}$   
 $(s.sid = r_1.sid \wedge r_1.sid = r_2.sid \wedge r_1.bid \neq r_2.bid$   
 $\wedge P.sname = s.sname) \}$

Q7) Find the names of sailors who have reserved all boats.

{ P |  $\exists s \in \text{sailors } \forall b \in \text{Boats } (\exists r \in \text{Reserves}$   
 $(s.sid = r.sid \wedge r.bid = b.bid \wedge P.sname = s.sname)) \}$

Q8) Find the sailors who have reserved all red Boats.

{ s |  $s \in \text{sailors} \wedge \forall b \in \text{Boats}$   
 $(b.color = 'red' \Rightarrow (\exists r \in \text{Reserves} (s.sid = r.sid$   
 $\wedge r.bid = b.bid))) \}$

Result:

Successfully written the Tuple Relational calculus for the given queries.

### \* Lab Exercise-7:

- Aim: To create the Supply-Information database with the following tables

Suppliers(sid:integer, sname:string, address:string)

Parts(pid:integer, pname:string, color:string)

catalog(sid:integer, pid:integer, cost:real)

and perform the given queries.

### • SQL Queries:

#### → creation of Database:

mysql > CREATE DATABASE Supply\_Information;

mysql > USE Supply\_Information;

#### → creation of Tables:

mysql > CREATE TABLE Suppliers(

- >     sid INT PRIMARY KEY,
- >     sname VARCHAR(50),
- >     address VARCHAR(100)
- > );

mysql > CREATE TABLE Parts(

- >     pid INT PRIMARY KEY,
- >     pname VARCHAR(50),
- >     color VARCHAR(10),
- > );

```
mysql> CREATE TABLE catalog (
    >     sid INT PRIMARY KEY,
    >     pid INT PRIMARY KEY,
    >     cost REAL,
    >     FOREIGN KEY(sid) REFERENCES
    >                           Suppliers (sid),
    >     FOREIGN KEY(pid) REFERENCES
    >                           Parts(Pid)
    > );
```

→ Inserting the values into Suppliers table

```
mysql> INSERT INTO Suppliers VALUES
    > (1,'Supplier A','123 Main st'),
    > (2,'Supplier B','456 OAK st'),
    > (3,'Supplier C','789 Maple st'),
    > (4,'Supplier D','221 Packer Ave');
```

```
mysql> SELECT * FROM Suppliers;
```

sid	name	address
1	Supplier A	123 Main st
2	Supplier B	456 OAK st
3	Supplier C	789 Maple st
4	Supplier D	221 Packer Ave

→ Inserting the values into Parts table

```
mysql> INSERT INTO Parts VALUES
    > (101, 'Part X', 'red'), (102, 'Part Y', 'green'),
    > (103, 'Part Z', 'blue'), (104, 'Part W', 'red'),
    > (105, 'Part V', 'green');
```

```
mysql> SELECT * FROM Parts;
```

Pid	Pname	color
101	Part X	red
102	Part Y	green
103	Part Z	blue
104	Part W	red
105	Part V	green

→ Inserting the values into catalog table

```
mysql> INSERT INTO catalog VALUES
    > (1, 101, 150.00), (1, 102, 250.00),
    > (2, 103, 120.00), (2, 101, 140.00),
    > (3, 104, 300.00), (4, 101, 190.00),
    > (3, 105, 200.00), (4, 102, 210.00),
    > (4, 104, 180.00), (4, 105, 195.00);
```

```
mysql> SELECT * FROM catalog;
```

sid	Pid	cost
1	101	150.00
1	102	250.00
2	103	120.00
2	101	140.00
3	104	300.00
3	105	200.00
4	101	190.00
4	102	210.00
4	104	180.00
4	105	195.00

Q1) Find the names of suppliers who supply some red part

```
mysql > SELECT DISTINCT S.sname
      → FROM Suppliers S
      → JOIN Catalog C ON S.sid=C.sid
      → JOIN Parts P ON P.pid=C.pid
      → WHERE P.color='red';
```

sname
Supplier A
Supplier B
Supplier C
Supplier D

Q2) Find the sids of suppliers who supply some red or green part.

```
mysql > SELECT DISTINCT C.sid
      → FROM Catalog C
      → JOIN Parts P ON P.pid=C.pid
      → WHERE P.color='red' OR
      → P.color='green';
```

sid
1
2
4
3

Q3) Find the sids of suppliers who supply some red part or at 221 Packer Ave.

```
mysql > SELECT DISTINCT c.sid
      → FROM Catalog C
      → JOIN Parts P ON P.pid = C.pid
      → JOIN Suppliers S ON S.sid = C.sid
      → WHERE P.color = 'red' OR
      → S.address = '221 Packer Ave';
```

sid
1
2
3
4

Q4) Find the sids of suppliers who supply some red part and some green part.

```
mysql > SELECT DISTINCT c.sid
      → FROM Catalog C
      → JOIN Parts P ON P.pid = C.pid
      → JOIN Catalog C1 ON C.sid = C1.sid
      → JOIN Parts P1 ON P1.pid = C1.pid
      → WHERE P.color = 'red' AND P1.color = 'green';
```

sid
4
1
3

Q5) Find the sids of suppliers who supply every part.

```
mysql > SELECT S.sid FROM Suppliers S
      → WHERE NOT EXISTS (
      →   SELECT P.pid FROM Parts P
      →   WHERE NOT EXISTS (
      →     SELECT C.pid
      →     FROM Catalog C
      →     WHERE C.sid = S.sid AND C.pid = P.pid
      →   )
      → );
```

Output:  
Empty set  
(0.00 sec)

Q6) Find the sids of suppliers who supply every red part.

```
mysql > SELECT s.sid FROM Suppliers s
      → WHERE NOT EXISTS (
      →   → SELECT p.pid FROM Parts P
      →   → WHERE P.color='red' AND
      →   → NOT EXISTS (
      →   →   → SELECT c.pid FROM catalog C
      →   →   → WHERE C.sid=s.sid AND C.pid=P.pid
      →   →   → )
      →   → );
```

sid
4

Q7) Find the sids of suppliers who supply every red or green part.

```
mysql > SELECT c.sid
      → FROM catalog C, Parts P
      → WHERE C.pid=P.pid AND
      → (P.color='red' OR P.color='green')
      → GROUP BY C.sid
      → HAVING COUNT(DISTINCT C.pid)=(SELECT COUNT(*) FROM Parts
      → WHERE color IN ('red', 'green'))
      → )
      → ;
```

sid
4

Q8) Find the sids of suppliers who supply every red part or supply every green part.

```
mysql> SELECT S.sid FROM Suppliers S
    → WHERE NOT EXISTS (
    →     SELECT P.pid FROM Parts P
    →     WHERE P.color = 'red' AND
    →     NOT EXISTS (
    →         SELECT C.pid FROM Catalog C
    →         WHERE C.sid = S.sid AND P.pid = C.pid
    →     ))
    → ) OR (NOT EXISTS (
    →     SELECT P.pid FROM Parts P
    →     WHERE P.color = 'green' AND
    →     NOT EXISTS (
    →         SELECT C.pid FROM catalog C
    →         WHERE C.sid = S.sid AND P.pid = C.pid
    →     ))
    → ) ;
```

sid
4

Q9) Find parts of sids such that the supplier with the first sid charges more for some part than the supplier with second sid.

```
mysql> SELECT C1.sid, C2.sid
    → FROM catalog C1, catalog C2
    → WHERE C1.pid = C2.pid AND
    → C1.cost > C2.cost
```

sid	sid
1	2
1	4
3	4
3	4
4	1
4	2

Q10) Find the pids of parts that are supplied by atleast two different suppliers.

```
mysql> SELECT P.pid
      → FROM Parts P, Catalog C
      → WHERE P.pid = C.pid
      → GROUP BY P.pid
      → HAVING COUNT(DISTINCT C.sid) >= 2;
```

Pid
101
102
104
105

Q11) Find the pids of the most expensive parts supplied by supplier named Yosemite sham( i.e supplier D)

```
mysql> SELECT pid FROM Catalog C
      → WHERE C.sid IN(
      →     SELECT sid FROM Suppliers
      →     WHERE sname = "Supplier D"
      → )
      → AND C.cost = (
      →     SELECT MAX(c2.cost) FROM Catalog C2
      →     WHERE C2.sid = C.sid
      → )
      → ;
```

Pid
102

Q12) Find the pids of parts supplied by every supplier at less than \$200.

```
mysql > SELECT P.pid FROM Parts P  
→ WHERE NOT EXISTS (  
→     SELECT S.sid FROM Suppliers S  
→     WHERE NOT EXISTS (  
→         SELECT C.pid FROM catalog C  
→         WHERE C.sid=S.sid AND C.pid=P.pid  
→         AND C.cost < 200  
→     )  
→ );
```

Output :

EMPTY SET (0.00 sec)

Result :

successfully created the Supply-Information database with tables (Suppliers, Parts, Catalog) and performed the given queries.