EXPERIMENT - 4

AIM: Implementation of various set operators in SQL.

THEORY: SQL provides set operators to compare rows from two or more tables or to combine the results obtained from two or more queries to obtain the final result. These operators are used to join the results of two (or more) SELECT statements. The set operators are UNION, UNION ALL, INTERSECT, INTERSECT ALL, and EXCEPT/MINUS.

a) UNION: It combines the result of two or more SELECT statements. To successfully execute the operation of UNION, the number of columns and the data type must be same in both tables. After performing the UNION operation, the duplicate rows will be eliminated from the results.

Syntax:
```
SELECT exp1, exp2 ... expn
FROM table1
WHERE condition
UNION
SELECT exp1, exp2 ... expn
FROM table2
WHERE condition;
```

Example:
```
SELECT * FROM empl_employees
UNION
```

SELECT * FROM compl_employees;

b) INTERSECT : It allows to find results that exist in both queries. The number of columns and data type must be same in both tables. After performing the INTERSECT operation, the data/records which are common in both the SELECT statements are returned.

Syntax: SELECT exp1, exp2, exp3... expN FROM table 1
WHERE condition
INTERSECT
SELECT exp1, exp2, exp3... expN FROM table 2
WHERE condition;

Example: SELECT * FROM compl_employees
INTERSECT
SELECT * FROM comp2 _employees;

c) Minus / EXCEPT : It allows to filter out the results which are present in the first query but absent in the second query. The number of columns and data types must be same in both tables. After performing the EXCEPT operation, the data / record which are not present in the second SELECT statement or query are displayed.

Syntax: SELECT exp1, exp2 ... expN FROM table 1 WHERE condition EXCEPT
SELECT exp1, exp2.. expN FROM table 2 WHERE condition;

Example : SELECT * FROM comp1_employees
EXCEPT
SELECT * FROM comp2_employees;

INPUT :

```
DROP TABLE IF EXISTS Sailors;
DROP TABLE IF EXISTS Boats;
DROP TABLE IF EXISTS Reserves;
CREATE TABLE Sailors (SID number(5),sname varchar2(10),rating
number(1.1),age number(3));
CREATE TABLE Boats (BID number(5),bname varchar2(10),color varchar2(10));
CREATE TABLE Reserves (SID number(5),BID number(5),day DATE);
INSERT INTO Sailors
VALUES(1,'Amar',4.9,41),(2,'Vinay',4.7,34),(3,'Rohit',4.5,35),(4,'Sachin',
4.8,49);
INSERT INTO Boats VALUES
(1007,'Boat1','Green'),(1018,'Boat2','Red'),(1045,'Boat3','Blue'),(1010,'B
oat4','Green'),(1035,'Boat5','Red');
INSERT INTO Reserves VALUES (1,1007,'7-July-2022'),(4,1010,'5-November-
2022'),(2,1045,'30-April-2022'),(1,1018,'24-April-2022'),(3,1035,'14-
March-2022');
SELECT * FROM Sailors;
SELECT * FROM Boats;
SELECT * FROM Reserves;
```

DATABASES :

Reserves

SID	BID	day
1	1007	7-July-2022
4	1010	5-November-2022
2	1045	30-April-2022
1	1018	24-April-2022
3	1035	14-March-2022

Boats

BID	bname	color
1007	Boat1	Green
1018	Boat2	Red
1045	Boat3	Blue
1010	Boat4	Green
1035	Boat5	Red

Sailors

SID	sname	rating	age
1	Amar	4.9	41
2	Vinay	4.7	34
3	Rohit	4.5	35
4	Sachin	4.8	49

Query 1 : Find Sailor ID's who have reserved red boats or green boats.

Code :

```
SELECT s.SID
FROM Sailors s
WHERE s.SID IN (SELECT r.SID FROM Reserves r,Boats b WHERE b.color='Red'
and b.BID=r.BID
UNION
SELECT r.SID FROM Reserves r,Boats b WHERE b.color='Green' and
b.BID=r.BID);
```

Output :

SID
1
3
4

Query 2 : Find Sailor names who have reserved both red and green boats.

Code :

```
SELECT s.sname
FROM Sailors s
WHERE s.SID IN (SELECT r.SID FROM Reserves r,Boats b WHERE b.color='Red'
and b.BID=r.BID
INTERSECT
SELECT r.SID FROM Reserves r,Boats b WHERE b.color='Green' and
b.BID=r.BID);
```

Output :

sname
Amar

Query 3 : Find Sailor names who have reserved red boats but not green boats.

Code :

```
SELECT s.sname
FROM Sailors s
WHERE s.SID IN (SELECT r.SID FROM Reserves r,Boats b WHERE b.color='Red'
and b.BID=r.BID
EXCEPT
SELECT r.SID FROM Reserves r,Boats b WHERE b.color='Green' and
b.BID=r.BID);
```

Output :

sname
Rohit

CONCLUSION: Various SET Operators (UNION, INTERSECT, EXCEPT) in SQL has been understood and executed in the queries.