Government College of Engineering, Jalgaon		
(An Autonomous Institute of Government of Maharashtra)		
Name :	**Semester :** V	**PRN :**
Class : T. Y. B.Tech Computer	**Academic Year :** 2024-25	**Subject :** CO307U DBMS Lab
Course Teacher : Mr. Vinit Kakde		
Date of Performance :	**Date of Completion :**	

Practical no. 8

Aim: Queries using any, all, in, intersect, union

Theory:

Any Operator:

The any operator is used in a WHERE or HAVING clause to compare a value with any of the values returned from a subquery. The **ANY** operator returns true if the comparison is true for any of the values in the subquery.

ALL Operator

SQL all compares a value of the first table with all values of the second table and returns the row if there is a match with all values.

IN Operator

- IN operator allows us to easily test if the expression matches any value in the list of values.
- It is used to remove the need for multiple **OR** conditions in SELECT, INSERT, UPDATE, or DELETE.
- We can also use **NOT IN** to exclude the rows in our list. We should note that any kind of duplicate entry will be retained.

Intersect operator

SQL INTERSECT operator combines two select statements and returns only the dataset that is common in both the statements. To put it simply, it acts as a mathematical intersection. In mathematics, the intersection of A and B is the common data present in both A and B. Thus, when you provide two select queries to combine, the SQL INTERSECT will only return the common rows from both the SELECT queries.

Union operator

The UNION operator is used to combine the data from the result of two or more SELECT command queries into a single distinct result set. This operator removes any duplicates present in the results being combined.

Queries and output:

1) **Syntax:**
   Create table table_name(column1 datatype, column2 datatype,….. column-N datatype);

   Query:
   Create table sailors(sid integer, sname varchar(20),rating integer, age integer);

```
mysql> desc sailors;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| sid    | int         | YES  |     | NULL    |       |
| sname  | varchar(20) | YES  |     | NULL    |       |
| rating | int         | YES  |     | NULL    |       |
| age    | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.70 sec)
```

2) **Syntax:**
   insert into orders values(&o_id,&orderno,&p_id);

   Query:
   Insert into sailors values(22,’dustin’,7,45);
   Insert into sailors values(29,’brutus’,1,33);
   Insert into sailors values(31,’lubber’,79,55);
   Insert into sailors values(32,’andy’,8,25);
   Insert into sailors values(58,’rusty’,10,35);
   Insert into sailors values(58,’buplb’,10,35);

Insert into sailors values(58,'buplerb',10,35);

```
mysql> select * from sailors;
+-------+----------+---------+-------+
| sid   | sname    | rating  | age   |
+-------+----------+---------+-------+
|    22 | dustin   |       7 |    45 |
|    29 | brutus   |       1 |    33 |
|    31 | lubber   |      79 |    55 |
|    32 | andy     |       8 |    25 |
|    58 | rusty    |      10 |    35 |
|    58 | buplb    |      10 |    35 |
|    58 | buplerb  |      10 |    35 |
+-------+----------+---------+-------+
7 rows in set (0.07 sec)
```

3) **Syntax:**
Create table table_name(column1 datatype, column2 datatype,….. column-N datatype);

Query:
Create table boats(bid integer, bname varchar(20),color varchar(20));

```
mysql> desc boats;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| bid   | int         | YES  |     | NULL    |       |
| bname | varchar(20) | YES  |     | NULL    |       |
| color | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

4) **Syntax:**
insert into orders values(&o_id,&orderno,&p_id);

Query:
Insert into boats values(101,'interlake','blue');
Insert into boats values(102,'interlake','red');
Insert into boats values(103,'clipper','green');

Insert into boats values(104,’marine’,’red’);

```
mysql> select * from boats;
+------+--------------+--------+
| bid  | bname        | color  |
+------+--------------+--------+
|  101 | interlake    | blue   |
|  102 | interlake    | red    |
|  103 | clipper      | green  |
|  104 | marine       | red    |
+------+--------------+--------+
4 rows in set (0.00 sec)
```

5) **Syntax:**
Create table table_name(column1 datatype, column2 datatype,….. column-N datatype);

Query:
Create table reserves(sid integer, bname integer,day date);

```
mysql> desc reserves;
+-------+------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+------+------+-----+---------+-------+
| sid   | int  | YES  |     | NULL    |       |
| bid   | int  | YES  |     | NULL    |       |
| day   | date | YES  |     | NULL    |       |
+-------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

6) **Syntax:**
insert into orders values(&o_id,&orderno,&p_id);

Query:
Insert into reserves values(22,101,’2004-01-01’);
Insert into reserves values(22,102,’2004-01-01’);
Insert into reserves values(22,103,’2004-02-01’);
Insert into reserves values(22,105,’2004-02-01’);
Insert into reserves values(31,103,’2005-05-05’);
Insert into reserves values(32,104,’2005-04-07’);

```
mysql> select * from reserves;
+------+------+------------+
| sid  | bid  | day        |
+------+------+------------+
|   22 |  101 | 2004-01-01 |
|   22 |  102 | 2004-01-01 |
|   22 |  103 | 2004-02-01 |
|   22 |  105 | 2004-02-01 |
|   31 |  103 | 2005-05-05 |
|   32 |  104 | 2005-04-07 |
+------+------+------------+
6 rows in set (0.00 sec)
```

7) Find all Sailors id's of sailors who have a rating of at least 8 or reserved boat 103

```
mysql> (select sid from sailors where rating>=8)union(select sid from reserves where bid=103);
+------+
| sid  |
+------+
|   31 |
|   32 |
|   58 |
|   22 |
+------+
4 rows in set (0.03 sec)
```

8) Find all Sailors id's of sailors who have a rating of at least 8 and reserved boat 103

```
mysql> (select sid from sailors where rating>=8)intersect(select sid from reserves where bid=103);
+------+
| sid  |
+------+
|   31 |
+------+
1 row in set (0.00 sec)
```

9) Find the names of sailors who have reserved boat number 103

```
mysql> select s.sname from sailors s where s.sid in(select r.sid from reserves r where r.bid=103);
+--------+
| sname  |
+--------+
| dustin |
| lubber |
+--------+
2 rows in set (0.00 sec)
```

10) Find the names of sailors who have never reserved boat number 103

```
mysql> select s.sname from sailors s where s.sid not in(select r.sid from reserves r where r.bid=103);
+---------+
| sname   |
+---------+
| brutus  |
| andy    |
| rusty   |
| buplb   |
| buplerb |
+---------+
5 rows in set (0.03 sec)
```

11) Find sailors whose rating is better than some sailor called Horatio

```
mysql> select s.sid from sailors s where s.rating>any(select s2.rating from sailors s2 where s2.sname='Horatio');
Empty set (0.03 sec)
```

12) Find the sailors with the highest rating

```
mysql> select s.sid from sailors s where s.rating>=any(select s2.rating from sailors s2);
+------+
| sid  |
+------+
|   22 |
|   29 |
|   31 |
|   32 |
|   58 |
|   58 |
|   58 |
+------+
7 rows in set (0.00 sec)
```

13) Find the age of the sailors whose name start with ‘b’ and ends with ‘b’

```
mysql> select age from sailors where sname like'b%b';
+------+
| age  |
+------+
|   35 |
|   35 |
+------+
2 rows in set (0.00 sec)
```

14) Find the names of sailors whose age is between 30 and 45

```
mysql> select sname from sailors where age between 30 and 45;
+---------+
| sname   |
+---------+
| dustin  |
| brutus  |
| rusty   |
| buplb   |
| buplerb |
+---------+
5 rows in set (0.00 sec)
```

15) Display the sorted list of sailors names

```
mysql> select sname from sailors order by sname;
+----------+
| sname    |
+----------+
| andy     |
| brutus   |
| buplb    |
| buplerb  |
| dustin   |
| lubber   |
| rusty    |
+----------+
7 rows in set (0.01 sec)
```

Conclusion:

SQL Operators are used to perform various operations on the data using SQL queries. These operators simplify arithmetic, comparison , logical, and bitwise operations on the data.

Questions and Answers :

1. **What is a PL/SQL procedure?**

Ans. A PL/SQL procedure is a named block of PL/SQL code that can be executed by calling it by name. Procedures can accept input parameters and return output parameters, but they do not return a value like a function.

2. **How do I create a PL/SQL procedure?**

Ans. To create a PL/SQL procedure, you can use the CREATE PROCEDURE statement. The syntax for creating a procedure is as follows:

```
CREATE PROCEDURE procedure_name (
  parameter1 datatype,
  parameter2 datatype,
  ...)
 AS
  -- PL/SQL code goes here
BEGIN
  -- PL/SQL code goes here
END procedure_name;
```

3.What is a PL/SQL function?

Ans. A PL/SQL function is a named block of PL/SQL code that can be executed by calling it by name. Functions can accept input parameters and return a value.

4.How do I create a PL/SQL function?

Ans. To create a PL/SQL function, you can use the CREATE FUNCTION statement. The syntax for creating a function is as follows:

```
CREATE FUNCTION function_name (
  parameter1 datatype,
  parameter2 datatype,
  ...)
 RETURN return_datatype AS
  -- PL/SQL code goes here
BEGIN
  -- PL/SQL code goes here
  RETURN return_value;
END function_name;
```

5.How do I call a PL/SQL procedure or function?

Ans. To call a PL/SQL procedure or function, you can use the EXECUTE or CALL statement. The syntax for calling a procedure or function is as follows:

EXECUTE procedure_name (parameter1, parameter2, ...);

CALL function_name (parameter1, parameter2, ...) INTO return_variable;

Name & Sign of Course Teacher

Mr. Vinit Kakde