PRACTICAL NO. 5

OBJECTIVE (AIM) OF THE EXPERIMENT

Querying the Database based on Set, Arithmetic(and Logical operator.(AND,OR,BETWEEN,NOT,LIKE, Addition,Multiplication,Subtraction,Division)

PROCEDURE

a) Procedure for doing the experiment:

Step no.	Details of the step
1	**Set Operators:** The Set operator combines the result of 2 queries into a single result. The following are the operators: ● Union ● Union all ● Intersect ● Minus
2	**The rules to which the set operators are strictly adhere to :** The queries which are related by the set operators should have a same number of column and column definition. Such query should not contain a type of long. Labels under which the result is displayed are those from the first select statement.

b) SQL commands:

Union: Returns all distinct rows selected by both the queries

Syntax:

Query1 Union Query2;

Union all: Returns all rows selected by either query including the duplicates.

Syntax:

Query1 Union all Query2;

Intersect: Returns rows selected that are common to both queries.

Syntax:

Query1 Intersect Query2;

Minus: Returns all distinct rows selected by the first query and are not by the second

Syntax:

Query1 minus Query2;

EXCEPT

EXCEPT clause in SQL Server is working as like MINUS operation in Oracle. EXCEPT query returns all rows which are in the first query but those are not returned in the second query.

c) **Queries**:

SQL Operators

The operators are symbols (and keywords) that are used to perform operations with values.

These operators are used with SQL clauses such as: `SELECT`, `WHERE`, `ON` etc.
The operators in SQL can be categorized as:

- Arithmetic operators
- Comparison operators
- Logical operators

SQL Arithmetic Operators

Arithmetic operators perform simple arithmetic operations such as addition, subtraction, multiplication etc.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Divide
%	Modulo (Remainder)

Addition Operator

```
-- returns new column named total_amount which is
-- 100 added to the amount field
SELECT item, amount, amount+100 AS total_amount
FROM Orders;
```

Run Code

Subtraction Operator

```
-- returns new column named offer_price which is
-- 20 subtracted to the amount field
SELECT item, amount, amount-20 AS offer_price
FROM Orders;
```
Run Code

Multiplication Operator

```
-- returns new column named total_amount  which is
-- 4 multiplied to the amount field
SELECT item, amount, amount*4 AS total_amount
FROM Orders;
```
Run Code

Division Operator

```
-- returns new column named half_amount which is
-- divided by 2 to the amount field
SELECT item, amount, amount/2 AS half_amount
FROM Orders;
```
Run Code

Modulo (Remainder) Operator

```
-- returns 1 which is remainder
SELECT 10 % 3 AS result;
```
Run Code

Consider the following Tables:

EMPLOYEE(Emp_id, EMP_name,Job_name,Manager_id,Hire_date,Salary,Deptno)

DEPARTMENT(Deptno, Dname, MGRSSN)

PROJECT(Pname,Pno,Plocation,Deptno)

emp_id	emp_name	job_name	manager_id	hire_date	salary	E_Bonus	dep_no
68319	KAYLING	PRESIDENT		1991-11-18	6000.00	300.00	1001
66928	BLAZE	MANAGER	68319	1991-05-01	2750.00	200.00	3001
67832	CLARE	MANAGER	68319	1991-06-09	2550.00	200.00	1001
65646	JONAS	MANAGER	68319	1991-04-02	2957.00	200.00	2001
67858	SCARLET	ANALYST	65646	1997-04-19	3100.00	250.00	2001
69062	FRANK	ANALYST	65646	1991-12-03	3100.00	250.00	2001
63679	SANDRINE	CLERK	69062	1990-12-18	900.00	150.00	2001
64989	ADELYN	SALESMAN	66928	1991-02-20	1700.00	180.00	3001
65271	WADE	SALESMAN	66928	1991-02-22	1350.00	180.00	3001
66564	MADDEN	SALESMAN	66928	1991-09-28	1350.00	180.00	3001
68454	TUCKER	SALESMAN	66928	1991-09-08	1600.00	180.00	3001
68736	ADNRES	CLERK	67858	1997-05-23	1200.00	150.00	2001

```
  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |   150.00   3001
  69324 | MARKER   | CLERK     |      67832 | 1992-01-23 | 1400.00 |   150.00   1001
```

Department Table

deptno	dname	Citylocation	dCountry
1001	Accounting	New York	United States of America,
2001	Research	Dallas	United States
3001	Sales	Chicago	United States of America
4001	Marketing	Los Angeles	United States

Project Table

Pno	Pname	PCitylocation	PCountry
111	P_1	New York	United States of America,
112	P_2	Dallas	United States
113	P_3	Chicago	United States of America
114	P_4	Denmark	northern Europe
115	P_5	Paris	France
116	P_6	Chicago	United States of America

Write a query for the following:-

Q1. Display all the Departments and Projects available.

```
mysql> SELECT deptno, dname AS name, Citylocation AS location, dCountry AS country FROM Department_160 UNION
    -> SELECT Pno, Pname AS name, PCitylocation AS location, PCountry AS country FROM Project_160 ;
+--------+------------+-------------+-------------------------+
| deptno | name       | location    | country                 |
+--------+------------+-------------+-------------------------+
|   1001 | Accounting | New York    | United States of America |
|   2001 | Research   | Dallas      | United States           |
|   3001 | Sales      | Chicago     | United States of America |
|   4001 | Marketing  | Los Angeles | United States           |
|    111 | P_1        | New York    | United States of America |
|    112 | P_2        | Dallas      | United States           |
|    113 | P_3        | Chicago     | United States of America |
|    114 | P_4        | Denmark     | Northern Europe         |
|    115 | P_5        | Paris       | France                  |
|    116 | P_6        | Chicago     | United States of America |
+--------+------------+-------------+-------------------------+
10 rows in set (0.00 sec)
```

Q2. Display the Locations of Departments and Projects.

```
mysql> SELECT Citylocation AS location FROM Department_160 UNION SELECT PCitylocation AS location FROM Project_160 ;
+-------------+
| location    |
+-------------+
| New York    |
| Dallas      |
| Chicago     |
| Los Angeles |
| Denmark     |
| Paris       |
+-------------+
6 rows in set (0.00 sec)
```

Q3. Display the Project's Locations which are not the Department's Locations.

```
mysql> SELECT DISTINCT PCitylocation AS location FROM Project_160
    -> WHERE PCitylocation NOT IN (SELECT Citylocation FROM Department_160 );
+----------+
| location |
+----------+
| Denmark  |
| Paris    |
+----------+
2 rows in set (0.01 sec)
```

Q4. Display the Department's Locations which are also Project's Locations.

```
mysql> SELECT DISTINCT Citylocation AS location FROM Department_P5_160
    -> WHERE Citylocation IN (SELECT PCitylocation FROM Project_P5_160);
+----------+
| location |
+----------+
| New York |
| Dallas   |
| Chicago  |
+----------+
3 rows in set (0.00 sec)
```

Q5. Display the cities of United States of America in which Projects are been designed and also display their respective Departments.

```
mysql> SELECT DP.Citylocation AS location, DP.dname AS department_name, PP.Pname AS project_name
    -> FROM Department_P5_160 DP
    -> JOIN Project_P5_160 PP ON DP.Citylocation = PP.PCitylocation
    -> WHERE DP.dCountry = 'United States of America' AND PP.PCountry = 'United States of America';
+----------+-----------------+--------------+
| location | department_name | project_name |
+----------+-----------------+--------------+
| New York | Accounting      | P_1          |
| Chicago  | Sales           | P_3          |
| Chicago  | Sales           | P_6          |
+----------+-----------------+--------------+
3 rows in set (0.00 sec)
```

Q6. Display the Countries and cities for projects P_1 and P_2 & Departments Accounting and Marketing.

```
mysql> SELECT Citylocation AS location, dCountry AS country FROM Department_P5_160
    -> WHERE dname IN ('Accounting', 'Marketing')
    -> UNION
    -> SELECT PCitylocation AS location, PCountry AS country FROM Project_P5_160
    -> WHERE Pname IN ('P_1', 'P_2');
+-------------+-------------------------+
| location    | country                 |
+-------------+-------------------------+
| New York    | United States of America |
| Los Angeles | United States           |
| Dallas      | United States           |
+-------------+-------------------------+
3 rows in set (0.00 sec)
```

Q7. Display those Cities which are same for Projects and Departments.

```
mysql> SELECT Citylocation AS location FROM Department_P5_160
    -> WHERE Citylocation IN (SELECT PCitylocation FROM Project_P5_160);
+----------+
| location |
+----------+
| New York |
| Dallas   |
| Chicago  |
+----------+
3 rows in set (0.00 sec)
```

Q8. Display Project numbers and Department numbers for which country is United States.

```
mysql> SELECT deptno AS department_number, NULL AS project_number FROM Department_160
    -> WHERE dCountry = 'United States'
    -> UNION
    -> SELECT NULL AS department_number, Pno AS project_number FROM Project_160
    -> WHERE PCountry = 'United States';
+-----------------+----------------+
| department_number | project_number |
+-----------------+----------------+
|              2001 |           NULL |
|              4001 |           NULL |
|              NULL |            112 |
+-----------------+----------------+
3 rows in set (0.00 sec)
```

Q9. Find the names of the projects and Departments which have city as Chicago.

```
mysql> SELECT dname AS department_name, NULL AS project_name FROM Department_160
    -> WHERE Citylocation = 'Chicago'
    -> UNION
    -> SELECT NULL AS department_name, Pname AS project_name FROM Project_160
    -> WHERE PCitylocation = 'Chicago';
+-----------------+--------------+
| department_name | project_name |
+-----------------+--------------+
| Sales           | NULL         |
| NULL            | P_3          |
| NULL            | P_6          |
+-----------------+--------------+
3 rows in set (0.00 sec)
```

Q10.Display the details for projects and Departments which don't have country as Northern Europe.

```
mysql> SELECT deptno AS department_number, dname AS name, Citylocation AS location, dCountry AS country
    -> FROM Department_160   WHERE dCountry != 'Northern Europe'
    -> UNION
    -> SELECT NULL AS department_number, Pname AS name, PCitylocation AS location, PCountry AS country
    -> FROM Project_160   WHERE PCountry != 'Northern Europe';
+-----------------+------------+-------------+-------------------------+
| department_number | name       | location    | country                 |
+-----------------+------------+-------------+-------------------------+
|            1001 | Accounting | New York    | United States of America |
|            2001 | Research   | Dallas      | United States           |
|            3001 | Sales      | Chicago     | United States of America |
|            4001 | Marketing  | Los Angeles | United States           |
|            NULL | P_1        | New York    | United States of America |
|            NULL | P_2        | Dallas      | United States           |
|            NULL | P_3        | Chicago     | United States of America |
|            NULL | P_5        | Paris       | France                  |
|            NULL | P_6        | Chicago     | United States of America |
+-----------------+------------+-------------+-------------------------+
9 rows in set (0.00 sec)
```

Q11 Get details of the Employee with the largest Salary.

```
mysql> SELECT * FROM Employee_160  WHERE Salary = (SELECT MAX(Salary) FROM Employee_160 );
+--------+----------+-----------+------------+------------+---------+---------+--------+
| Emp_id | Emp_name | Job_name  | Manager_id | hire_date  | Salary  | E_bonus | dep_no |
+--------+----------+-----------+------------+------------+---------+---------+--------+
|  68319 | KAYLING  | PRESIDENT |      11001 | 1991-11-18 | 6000.00 |  300.00 |   1001 |
+--------+----------+-----------+------------+------------+---------+---------+--------+
1 row in set (0.00 sec)
```

Q12. Display the Total Salary of Employees including Bonus.

```
mysql> SELECT Emp_id, Emp_name, (Salary + E_bonus) AS Total_Salary FROM Employee_160  ;
+--------+----------+--------------+
| Emp_id | Emp_name | Total_Salary |
+--------+----------+--------------+
|  68319 | KAYLING  |      6300.00 |
|  66928 | BLAZE    |      2950.00 |
|  67832 | CLARE    |      2750.00 |
|  65646 | JONAS    |      3157.00 |
|  67858 | SCARLET  |      3350.00 |
|  69062 | FRANK    |      3350.00 |
|  63679 | SANDRINE |      1050.00 |
|  64989 | ADELYN   |      1880.00 |
|  65271 | WADE     |      1530.00 |
|  66564 | MADDEN   |      1530.00 |
|  68454 | TUCKER   |      1780.00 |
|  68736 | ADNRES   |      1350.00 |
|  69000 | JULIUS   |      1200.00 |
|  69324 | MARKER   |      1550.00 |
+--------+----------+--------------+
14 rows in set (0.00 sec)
```

Q13. Display the Salaries if it is increased by 3 times more than original Salaries of Employees who work as Analyst.

```
mysql> SELECT Emp_name, Job_name, salary, CASE WHEN Job_name = 'ANALYST' THEN Salary * 3 ELSE Salary END AS Increased_Salary FROM Employee_160  WHERE Job_name = 'ANALYST';
+----------+----------+---------+-----------------+
| Emp_name | Job_name | salary  | Increased_Salary |
+----------+----------+---------+-----------------+
| SCARLET  | ANALYST  | 3100.00 |         9300.00 |
| FRANK    | ANALYST  | 3100.00 |         9300.00 |
+----------+----------+---------+-----------------+
2 rows in set (0.00 sec)
```

Q14. Display the Salaries of all Employees who are paying 10 % of their total salary for Social Cause.

```
mysql> SELECT Emp_id, Emp_name, Salary, E_bonus, CASE WHEN (Salary + E_bonus) * 0.10 > 0 THEN (Salary + E_bonus) * 0.10 ELSE 0 END AS Social_Contribution FROM Employee_160  ;
+--------+----------+---------+---------+--------------------+
| Emp_id | Emp_name | Salary  | E_bonus | Social_Contribution |
+--------+----------+---------+---------+--------------------+
|  68319 | KAYLING  | 6000.00 |  300.00 |           630.0000 |
|  66928 | BLAZE    | 2750.00 |  200.00 |           295.0000 |
|  67832 | CLARE    | 2550.00 |  200.00 |           275.0000 |
|  65646 | JONAS    | 2957.00 |  200.00 |           315.7000 |
|  67858 | SCARLET  | 3100.00 |  250.00 |           335.0000 |
|  69062 | FRANK    | 3100.00 |  250.00 |           335.0000 |
|  63679 | SANDRINE |  900.00 |  150.00 |           105.0000 |
|  64989 | ADELYN   | 1700.00 |  180.00 |           188.0000 |
|  65271 | WADE     | 1350.00 |  180.00 |           153.0000 |
|  66564 | MADDEN   | 1350.00 |  180.00 |           153.0000 |
|  68454 | TUCKER   | 1600.00 |  180.00 |           178.0000 |
|  68736 | ADNRES   | 1200.00 |  150.00 |           135.0000 |
|  69000 | JULIUS   | 1050.00 |  150.00 |           120.0000 |
|  69324 | MARKER   | 1400.00 |  150.00 |           155.0000 |
+--------+----------+---------+---------+--------------------+
14 rows in set (0.00 sec)
```