Objective:

Practice retrieving complex data sets that span across multiple related tables, allowing for better understanding of multi-dimensional data analysis.

Joins: Join, Left Outer Join, Right Outer Join, Self-Join

Consider the following database tables and write the solution for the given queries.

Tables: Employee (eid, ename, salary, doj, comm, did)

Department (did, departmentname, location)

Sample Data in Employees Table:

Q1) Problem Statement: Write a query to retrieve the names of employees along with their corresponding department names.

```
mysql> SELECT
    ->     e.ename AS Name,
    ->     d.departmentname AS DepartmentName
    -> FROM
    ->     Employee e
    -> JOIN
    ->     Department d
    -> ON
    ->     e.did = d.did;
+-----------------+----------------+
| Name            | DepartmentName |
+-----------------+----------------+
| Jim Halpert     | HR             |
| Stanley Hudson  | Finance        |
| Phyllis Vance   | IT             |
| Angela Martin   | HR             |
| Kevin Malone    | Finance        |
| Meredith Palmer | IT             |
| Andy Bernard    | HR             |
| Darryl Philbin  | Finance        |
| Oscar Martinez  | IT             |
| Toby Flenderson | HR             |
| Jane Smith      | Finance        |
| Alice Brown     | HR             |
| Bob Davis       | IT             |
| Carol White     | Marketing      |
+-----------------+----------------+
14 rows in set (0.00 sec)
```

Q2) Problem Statement: Write a query to find employees who work in departments located in either New York or London and have a salary greater than 60,000. Display their names, department names, locations, and salaries.

```
mysql> SELECT e.ename AS EmployeeName, d.departmentname AS DepartmentName, d.location AS Location, e.salary AS Salary FROM Employee e JOIN Department d ON e
.did = d.did WHERE d.location IN ('New York', 'London') AND e.salary > 60000;
+--------------+----------------+----------+--------+
| EmployeeName | DepartmentName | Location | Salary |
+--------------+----------------+----------+--------+
| Alice Brown  | HR             | New York |  75000 |
+--------------+----------------+----------+--------+
1 row in set (0.01 sec)
```

Q3)Problem Statement: Write a query to retrieve all employees and their corresponding department names, even if the employee is not assigned to any department.

```
mysql> SELECT
    ->     e.ename AS Name,
    ->     d.departmentname AS DepartmentName
    -> FROM
    ->     Employee e
    -> LEFT JOIN
    ->     Department d
    -> ON
    ->     e.did = d.did;
+-----------------+----------------+
| Name            | DepartmentName |
+-----------------+----------------+
| Jim Halpert     | HR             |
| Stanley Hudson  | Finance        |
| Phyllis Vance   | IT             |
| Angela Martin   | HR             |
| Kevin Malone    | Finance        |
| Meredith Palmer | IT             |
| Andy Bernard    | HR             |
| Darryl Philbin  | Finance        |
| Oscar Martinez  | IT             |
| Toby Flenderson | HR             |
| Jane Smith      | Finance        |
| Alice Brown     | HR             |
| Bob Davis       | IT             |
| Carol White     | Marketing      |
+-----------------+----------------+
14 rows in set (0.01 sec)
```

Q4) Problem Statement: Write a query to retrieve the names of all employees, their department names, and the location of their departments. Also, include employees who are not assigned to any department.

```
mysql> SELECT
    ->     e.ename AS EmployeeName,
    ->     d.departmentname AS DepartmentName,
    ->     d.location AS Location
    -> FROM
    ->     Employee e
    -> LEFT JOIN
    ->     Department d
    -> ON
    ->     e.did = d.did;
+-----------------+----------------+---------------+
| EmployeeName    | DepartmentName | Location      |
+-----------------+----------------+---------------+
| Jim Halpert     | HR             | New York      |
| Stanley Hudson  | Finance        | London        |
| Phyllis Vance   | IT             | San Francisco |
| Angela Martin   | HR             | New York      |
| Kevin Malone    | Finance        | London        |
| Meredith Palmer | IT             | San Francisco |
| Andy Bernard    | HR             | New York      |
| Darryl Philbin  | Finance        | London        |
| Oscar Martinez  | IT             | San Francisco |
| Toby Flenderson | HR             | New York      |
| Jane Smith      | Finance        | London        |
| Alice Brown     | HR             | New York      |
| Bob Davis       | IT             | San Francisco |
| Carol White     | Marketing      | Chicago       |
+-----------------+----------------+---------------+
14 rows in set (0.00 sec)
```

Q5) Problem Statement: Write a query to retrieve all departments and their corresponding employees, even if a department has no employees assigned to it.

```
mysql> SELECT
    ->     e.ename AS Name,
    ->     d.departmentname AS DepartmentName
    -> FROM
    ->     Employee e
    -> RIGHT JOIN
    ->     Department d
    -> ON
    ->     e.did = d.did;
+-----------------+----------------+
| Name            | DepartmentName |
+-----------------+----------------+
| Alice Brown     | HR             |
| Toby Flenderson | HR             |
| Andy Bernard    | HR             |
| Angela Martin   | HR             |
| Jim Halpert     | HR             |
| Jane Smith      | Finance        |
| Darryl Philbin  | Finance        |
| Kevin Malone    | Finance        |
| Stanley Hudson  | Finance        |
| Bob Davis       | IT             |
| Oscar Martinez  | IT             |
| Meredith Palmer | IT             |
| Phyllis Vance   | IT             |
| Carol White     | Marketing      |
+-----------------+----------------+
14 rows in set (0.00 sec)
```

Q6) Problem Statement: Write a query to retrieve all departments along with the names of employees assigned to them. If no employees are assigned, display "No Employees" in the employee name field. Also, include the department's location.

```
   d.departmen' at line 1
mysql> SELECT COALESCE(e.ename, 'No Employees') AS EmployeeName, d.departmentname AS DepartmentName, d.location AS Location FROM Department d LEFT JOIN Empl
oyee e ON d.did = e.did;
+----------------+----------------+--------------+
| EmployeeName   | DepartmentName | Location     |
+----------------+----------------+--------------+
| Alice Brown    | HR             | New York     |
| Toby Flenderson | HR             | New York     |
| Andy Bernard   | HR             | New York     |
| Angela Martin  | HR             | New York     |
| Jim Halpert    | HR             | New York     |
| Jane Smith     | Finance        | London       |
| Darryl Philbin | Finance        | London       |
| Kevin Malone   | Finance        | London       |
| Stanley Hudson | Finance        | London       |
| Bob Davis      | IT             | San Francisco |
| Oscar Martinez | IT             | San Francisco |
| Meredith Palmer | IT             | San Francisco |
| Phyllis Vance  | IT             | San Francisco |
| Carol White    | Marketing      | Chicago      |
+----------------+----------------+--------------+
14 rows in set (0.01 sec)
```

Q7) Problem Statement: Write a query to find pairs of employees who work in the same department.

```
mysql> SELECT
    ->     e1.ename AS Employee1,
    ->     e2.ename AS Employee2,
    ->     e1.did AS DepartmentID
    -> FROM
    ->     Employee e1
    -> JOIN
    ->     Employee e2
    -> ON
    ->     e1.did = e2.did
    -> AND
    ->     e1.eid < e2.eid;
+-----------------+-----------------+--------------+
| Employee1       | Employee2       | DepartmentID |
+-----------------+-----------------+--------------+
| Jim Halpert     | Angela Martin   |           10 |
| Jim Halpert     | Andy Bernard    |           10 |
| Jim Halpert     | Toby Flenderson |           10 |
| Jim Halpert     | Alice Brown     |           10 |
| Stanley Hudson  | Kevin Malone    |           20 |
| Stanley Hudson  | Darryl Philbin  |           20 |
| Stanley Hudson  | Jane Smith      |           20 |
| Phyllis Vance   | Meredith Palmer |           30 |
| Phyllis Vance   | Oscar Martinez  |           30 |
| Phyllis Vance   | Bob Davis       |           30 |
| Angela Martin   | Andy Bernard    |           10 |
| Angela Martin   | Toby Flenderson |           10 |
| Angela Martin   | Alice Brown     |           10 |
| Kevin Malone    | Darryl Philbin  |           20 |
| Kevin Malone    | Jane Smith      |           20 |
| Meredith Palmer | Oscar Martinez  |           30 |
| Meredith Palmer | Bob Davis       |           30 |
| Andy Bernard    | Toby Flenderson |           10 |
| Andy Bernard    | Alice Brown     |           10 |
| Darryl Philbin  | Jane Smith      |           20 |
| Oscar Martinez  | Bob Davis       |           30 |
| Toby Flenderson | Alice Brown     |           10 |
+-----------------+-----------------+--------------+
```

Q8) Problem Statement: Write a query to display employees names, working DepartmentName and their salary grade

```
mysql> SELECT
    ->     e.ename AS EmployeeName,
    ->     d.departmentname AS DepartmentName,
    ->     s.grade
    -> FROM
    ->     Employee e
    -> JOIN
    ->     Department d
    -> ON
    ->     e.did = d.did
    -> LEFT JOIN
    ->     Salgrade s
    -> ON
    ->     e.salary BETWEEN s.losal AND s.hisal;
+-----------------+-----------------+-------+
| EmployeeName    | DepartmentName  | grade |
+-----------------+-----------------+-------+
| Jim Halpert     | HR              |     5 |
| Stanley Hudson  | Finance         |     1 |
| Phyllis Vance   | IT              |     2 |
| Angela Martin   | HR              |     2 |
| Kevin Malone    | Finance         |     4 |
| Meredith Palmer | IT              |     2 |
| Andy Bernard    | HR              |     4 |
| Darryl Philbin  | Finance         |     3 |
| Oscar Martinez  | IT              |     3 |
| Toby Flenderson | HR              |     4 |
| Jane Smith      | Finance         |     3 |
| Alice Brown     | HR              |     5 |
| Bob Davis       | IT              |     5 |
| Carol White     | Marketing       |     5 |
+-----------------+-----------------+-------+
14 rows in set (0.01 sec)
```

Q9) Problem Statement: Write a query to calculate the average salary of employees in each department and list the department names and locations along with this average salary. Include only those departments that have more than one employee.

```
mysql>
mysql>
mysql> SELECT
    ->     d.departmentname AS DepartmentName,
    ->     d.location AS Location,
    ->     AVG(e.salary) AS AverageSalary
    -> FROM
    ->     Employee e
    -> JOIN
    ->     Department d
    -> ON
    ->     e.did = d.did
    -> GROUP BY
    ->     d.departmentname, d.location
    -> HAVING
    ->     COUNT(e.eid) > 1;
+----------------+---------------+---------------+
| DepartmentName | Location      | AverageSalary |
+----------------+---------------+---------------+
| HR             | New York      |    44800.0000 |
| Finance        | London        |    26800.0000 |
| IT             | San Francisco |    29000.0000 |
+----------------+---------------+---------------+
3 rows in set (0.01 sec)
```