Single Row Functions Assignment

1. Retrieve the employee names in uppercase along with their jobs from the EMP table.

```
Select upper(empname) as EmployeeNames,
job
From emp;
```

2. Fetch the employee names and their salaries. Round off the salary to 1 decimal place.

```
Select empname,round(sal,1) as
roundedSalary
From emp;
```

3. Find the employee names and the square root of their salaries. Display results for employees with salary more than 1000.

```
Select empname, sqrt(sal) as sqrtSalary
From emp
Where sal > 1000;
```

4. Show the employee names and salaries formatted with two decimal places.

```
Select empname, format(sal,2) as
formatted_salary
From emp;
```

5. Get the first 3 characters of all employee names.

```
Select substr(empname,1,3) as
formated_name
From emp;
```

6. Find the length of each employee's name and display it.

```
Select empname, length(empname) as
lengthOfName
From emp;
```

7. Display the names of employees and their salaries, but only for those whose salary is greater than 1500 and less than or equal to 3000.

```
Select empname, sal
From emp
Where sal > 1500 AND sal <=3000;
```

Subqueries Assignment

Get the names of employees who earn more than the average salary of the entire company.

- First, calculate the average salary of all employees
- Then, compare each employee's salary to the calculated average salary.

```
Select Empname
From emp
Where sal > (
    Select avg(sal)
    From emp
);
```

Find the employees who work in the same department as the employee 'JONES'.

- First, identify the deptno where 'JONES' is working
- Then, Select the names of employees who work in that department.

```
SELECT empname
From emp
Where deptno = (
    Select deptno
    From emp
    Where empname = 'JONES'
);
```

Retrieve the names of employees whose salaries are higher than the salary of 'BLAKE'.

- First, get the salary of BLAKE
- Then, compare other employees' salaries with this value and display the list of names who earn more.

```
Select empname
From emp
Where sal > (
    Select sal
    From emp
    Where empname=' BLAKE'
);
```

Display the names and hire dates of employees who were hired after 'SCOTT'.

- First, Get the hiredate of 'scott'.
- Compare this date with other employee's hiredates to list those who hired later.

```
select empname, hiredate
from emp
where hiredate > (
    select hiredate
    from emp
    where empname=' SCOTT'
) ;
```

Find the employees who do not have any manager (i.e., where MGR is NULL).

Without subquery:

```
Select empname
From emp
Where mgr is null;
```

With subquery:

```
Select empname
From emp
Where empno IN (
    Select empno
    From emp
    Where MGR is null
);
```

List the names of employees whose hire date is the same as any employee in department 30.

- Get the hiredates of employees in department 30
- Then, select the names of employees whose hiredate matches any of those hiredates which we will be getting with subquery.

```
SELECT empname
From emp
Where hiredate IN (
     Select hiredate
From emp
Where deptno = 30
);
```

JOINS

It allows us to retrieve data from multiple tables based on related columns.

1. INNER JOIN

Returns only where there is match in both the tables.

SYNTAX:

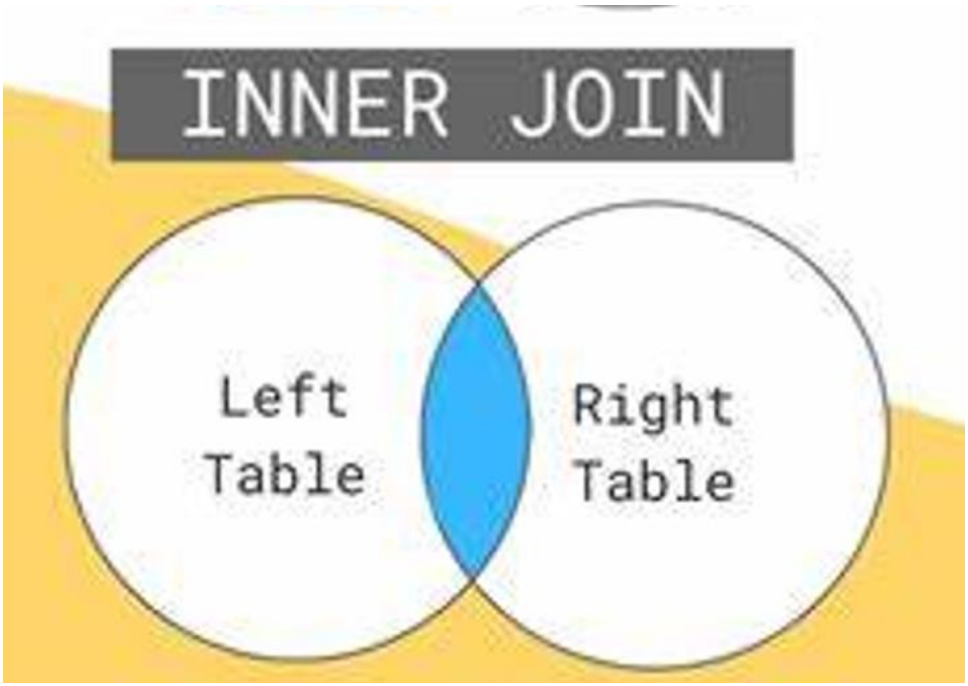
```
SELECT columns
From table1
Inner Join table2
ON table1.column = table2.column;
```

Ex. To retrieve employees and their department names.

```
Select emp.empname, emp.job, dept.dname
From emp
Inner join dept
ON emp.deptno = dept.deptno;
```

```
mysql> Select emp.empname, emp.job, dept.dname
    -> From emp
    -> Inner join dept
    -> ON emp.deptno = dept.deptno;
```

empname	job	dname
SMITH	SENIOR CLERK	RESEARCH
ALLEN	SALESMAN	SALES
WARD	SALESMAN	SALES
JONES	MANAGER	RESEARCH
MARTIN	SALESMAN	SALES
BLAKE	MANAGER	SALES
CLARK	MANAGER	ACCOUNTING
SCOTT	ANALYST	RESEARCH
KING	PRESIDENT	ACCOUNTING
TURNER	SALESMAN	SALES
ADAMS	CLERK	RESEARCH
JAMES	CLERK	SALES
FORD	ANALYST	RESEARCH
MILLER	CLERK	ACCOUNTING



2. Left Join(Left Outer Join)

It returns all rows from the left table and matched rows from the right table

Unmatched rows from the right table will display NULL.

SYNTAX:

```
Select columns
From table1
LEFT JOIN table2
ON table1.column = table2.column
```

Ex. To retrieve all employees and their department names, even if some employees do not belong to any department.

```
Select emp.empname,emp.job, dept.dname
From emp
Left join dept
On emp.deptno = dept.deptno;
```

TASK: add an employee who is working in deptno(60).
Then execute the query
And analyse.

3. Right Join

It returns all rows from the right table and matched rows from the left table.

Unmatched rows from the left table will display null values.

SYNTAX:

```
SELECT columns
From table1
RIGHT JOIN table2
ON table1.column = table2.column;
```

Ex. To retrieve all departments and their employees, including departments with no employees.

```
Select emp.empname, emp.job, dept.dname
From emp
Right join dept
On emp.deptno = dept.deptno;
```

```
mysql> Select emp.empname, emp.job, dept.dname
    -> From emp
    -> Right join dept
    -> On emp.deptno = dept.deptno;
+---------+--------------+------------+
| empname | job          | dname      |
+---------+--------------+------------+
| MILLER  | CLERK        | ACCOUNTING |
| KING    | PRESIDENT    | ACCOUNTING |
| CLARK   | MANAGER      | ACCOUNTING |
| FORD    | ANALYST      | RESEARCH   |
| ADAMS   | CLERK        | RESEARCH   |
| SCOTT   | ANALYST      | RESEARCH   |
| JONES   | MANAGER      | RESEARCH   |
| SMITH   | SENIOR CLERK | RESEARCH   |
| JAMES   | CLERK        | SALES      |
| TURNER  | SALESMAN     | SALES      |
```

```
| BLAKE   | MANAGER         | SALES       |
| MARTIN  | SALESMAN        | SALES       |
| WARD    | SALESMAN        | SALES       |
| ALLEN   | SALESMAN        | SALES       |
| NULL    | NULL            | OPERATIONS  |
+---------+-----------------+-------------+
```