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

- Select upper(empname) as Employee_Name, job
  From employee.

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

- Select empname, round(sal,1) as rounded_salary
  From employee;

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 salary_sqrt
  From employee
  Where sal > 1000;

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

- Select empname, format(sal, 2) as formatted_salary
  From employee;

5. Get the first 3 characters of all employee names

- Select substr(empname, 1, 3) as name_pre
  From employee;

6. Display employee names with leading spaces removed and trailing periods added (e.g., 'ALLEN' should become 'ALLEN.').

Select trim(empname) || '.' As modified_name
From employee;
SELECT CONCAT(TRIM(ENAME),'.') AS EMP_LIST FROM EMP;

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

- Select empname, length(empname) as 'name length'
  From employee;

8. 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 employee
Where sal> 1500 AND sal <= 3000;

JOINS

- To create department table

```
create table  dept (
    DEPTNO INT(2) PRIMARY KEY,
    DNAME VARCHAR(20),
    LOC VARCHAR(15)
);
```

- To insert values into table

```
INSERT INTO DEPT VALUES(10, 'ACCOUNTING', 'NEW YORK');
INSERT INTO DEPT VALUES(20, 'RESEARCH', 'DALLAS');
INSERT INTO DEPT VALUES(30, 'SALES', 'CHICAGO');
INSERT INTO DEPT VALUES(40, 'OPERATIONS', 'BOSTON');
```

JOINS:

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

Types of JOINS:

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 employee.empname, employee.job, dept.dname
From employee
Inner join dept
On employee.deptno = dept.deptno;
```

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 dnames, even if some employees do not belong to any department:

```
SELECT employee.empname, employee.job, dept.dname
From employee
Left join dept
On employee.deptno = dept.deptno;
```

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

3. Right Join

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

Unmatched rows from the left table will display the 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 employee.empname, employee.job, dept.dname
From employee
Right join dept
On employee.deptno = dept.deptno;
```

4. FULL JOIN (Full Outer Join)
   Returns all rows from both the table, with NULL in place of unmatched rows from either table.
   NOTE: MySQL does not support FULL OUTER JOIN directly but can be achieved using an UNION.

   STEPS:
   1. Perform a LEFT JOIN
   2. perform a right join
   3. combine results using UNION

   EX. To retrieve all department names and job titles;

```
Select dname as name
From dept
Union
Select job as jname
From employee;
```

- Write a query using INNER JOIN to display the employee name, job, salary, and department name for all employees.
- Write a query using LEFT JOIN to display the employee name, job, and department name for all employees, including those who may not belong to any department.
- Write a query using RIGHT JOIN to display all departments and the employees working in them, including departments that have no employees assigned.