```
create schema employee ;
use employee ;

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT AS DEPARTMENT
FROM emp_record_table;



-- EMP_RATING less than two
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING < 2;

-- EMP_RATING greater than four
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING > 4;


-- EMP_RATING between two and four
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING >= 2 AND EMP_RATING <= 4;


SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME
FROM emp_record_table
WHERE DEPT = 'Finance';


SELECT e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.GENDER, e.DEPT AS DEPARTMENT,
    COUNT(*) AS NUM_REPORTERS
FROM emp_record_table e
JOIN emp_record_table r ON e.EMP_ID = r.MANAGER_ID
```

```sql
GROUP BY e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.GENDER, e.DEPT
HAVING COUNT(*) > 0;



-- Query to list employees from the healthcare department
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT AS DEPARTMENT
FROM emp_record_table
WHERE DEPT = 'healthcare'
UNION
-- Query to list employees from the finance department
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT AS DEPARTMENT
FROM emp_record_table
WHERE DEPT = 'finance';


SELECT
    e.EMP_ID,
    e.FIRST_NAME,
    e.LAST_NAME,
    e.ROLE,
    e.DEPT AS DEPARTMENT,
    e.EMP_RATING,
    max_ratings.max_emp_rating AS MAX_EMP_RATING_FOR_DEPT
FROM
    emp_record_table e
JOIN (
    SELECT DEPT, MAX(EMP_RATING) AS max_emp_rating
    FROM emp_record_table
    GROUP BY DEPT
) max_ratings ON e.DEPT = max_ratings.DEPT
ORDER BY e.DEPT, e.EMP_ID;



SELECT ROLE, MIN(SALARY) AS MIN_SALARY, MAX(SALARY) AS MAX_SALARY
FROM emp_record_table
```

```
GROUP BY ROLE;
```

```
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    GENDER,
    DEPT AS DEPARTMENT,
    EXP,
    RANK() OVER (ORDER BY EXP DESC) AS EXPERIENCE_RANK
FROM
    emp_record_table;
```

```
CREATE VIEW high_salary_employees_view AS
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    GENDER,
    DEPT AS DEPARTMENT,
    COUNTRY,
    SALARY
FROM
    emp_record_table
WHERE
    SALARY > 6000;
    SELECT * FROM high_salary_employees_view;
```

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP
FROM emp_record_table
WHERE EXP > 10;
```

```sql
DELIMITER //
CREATE PROCEDURE GetEmployeesWithExperience()
BEGIN
   SELECT *
   FROM emp_record_table
   WHERE EXP > 3;
END //
DELIMITER ;

CALL GetEmployeesWithExperience();



USE employee;

SHOW TABLES;

-- Step 1: Check the execution plan
EXPLAIN SELECT *
FROM emp_record_table
WHERE FIRST_NAME = 'Eric';



-- Step 2: Create the index
ALTER TABLE emp_record_table
MODIFY COLUMN FIRST_NAME VARCHAR(255);

CREATE INDEX idx_first_name ON emp_record_table (FIRST_NAME(50));

-- Step 3: Verify index creation
SHOW INDEX FROM emp_record_table;
```

```
-- Step 4: Re-check the execution plan
EXPLAIN SELECT *
FROM emp_record_table
WHERE FIRST_NAME = 'Eric';


SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    SALARY,
    EMP_RATING,
    0.05 * SALARY * EMP_RATING AS BONUS
FROM
    emp_record_table;


    SELECT
    CONTINENT,
    COUNTRY,
    AVG(SALARY) AS AVG_SALARY
FROM
    emp_record_table
GROUP BY
    CONTINENT,
    COUNTRY;


DELIMITER //

CREATE FUNCTION GetJobProfile(EXP INT) RETURNS VARCHAR(50) DETERMINISTIC
BEGIN
    DECLARE job_profile VARCHAR(50);
    CASE
        WHEN EXP <= 2 THEN
            SET job_profile = 'JUNIOR DATA SCIENTIST';
        WHEN EXP <= 5 THEN
            SET job_profile = 'ASSOCIATE DATA SCIENTIST';
        WHEN EXP <= 10 THEN
            SET job_profile = 'SENIOR DATA SCIENTIST';
        WHEN EXP <= 12 THEN
            SET job_profile = 'LEAD DATA SCIENTIST';
        ELSE
            SET job_profile = 'MANAGER';
    END CASE;

    RETURN job_profile;
END //

DELIMITER ;

SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, GetJobProfile(EXP) AS JOB_PROFILE
FROM emp_record_table;
```

```
SELECT
    E1.EMP_ID,
    E1.FIRST_NAME,
    E1.LAST_NAME,
    E1.DEPARTMENT,
    COUNT(E2.EMP_ID) AS NUM_REPORTERS
FROM
    emp_record_table E1
LEFT JOIN
    emp_record_table E2 ON E1.EMP_ID = E2.MANAGER_ID
```

```
GROUP BY
    E1.EMP_ID, E1.FIRST_NAME, E1.LAST_NAME, E1.DEPARTMENT
HAVING
    NUM_REPORTERS > 0
ORDER BY
    E1.EMP_ID;


SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
FROM emp_record_table;


SELECT
    E1.EMP_ID,
    E1.FIRST_NAME,
    E1.LAST_NAME,
    COUNT(E2.EMP_ID) AS NUM_REPORTERS
FROM
    emp_record_table E1
LEFT JOIN
    emp_record_table E2 ON E1.EMP_ID = E2.MANAGER_ID
GROUP BY
    E1.EMP_ID, E1.FIRST_NAME, E1.LAST_NAME
HAVING
    NUM_REPORTERS > 0
ORDER BY
    E1.EMP_ID;
```