```
use employee_database;

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

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

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

-- Employees with EMP_RATING between two and four (inclusive)
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING BETWEEN 2 AND 4;


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


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

```
-- Employees from the Healthcare department
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT
FROM emp_record_table
WHERE DEPT = 'Healthcare'

UNION

-- Employees from the Finance department
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT
FROM emp_record_table
WHERE DEPT = 'Finance';



SELECT
    e.EMP_ID,
    e.FIRST_NAME,
    e.LAST_NAME,
    e.ROLE,
    e.DEPT,
    e.EMP_RATING,
    m.max_emp_rating
FROM
    emp_record_table e
INNER JOIN (
    SELECT DEPT, MAX(EMP_RATING) AS max_emp_rating
    FROM emp_record_table
    GROUP BY DEPT
) m ON e.DEPT = m.DEPT;
```

```
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,
   ROLE,
   DEPT,
   EXP,
   RANK() OVER (PARTITION BY DEPT ORDER BY EXP DESC) AS EXPERIENCE_RANK
FROM
   emp_record_table;



CREATE VIEW high_salary_employees AS
SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY
FROM emp_record_table
WHERE SALARY > 6000;

SELECT * FROM high_salary_employees;



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

```
DELIMITER //
CREATE PROCEDURE GetEmployeesWithExperience()
BEGIN
   SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP
   FROM emp_record_table
   WHERE EXP > 3;
END //
DELIMITER ;

CALL GetEmployeesWithExperience();



DELIMITER //

CREATE FUNCTION GetJobProfile(experience INT) RETURNS VARCHAR(50)
DETERMINISTIC
READS SQL DATA
BEGIN
   DECLARE job_profile VARCHAR(50);

   IF experience <= 2 THEN
      SET job_profile = 'JUNIOR DATA SCIENTIST';
   ELSEIF experience <= 5 THEN
      SET job_profile = 'ASSOCIATE DATA SCIENTIST';
   ELSEIF experience <= 10 THEN
      SET job_profile = 'SENIOR DATA SCIENTIST';
   ELSEIF experience <= 12 THEN
      SET job_profile = 'LEAD DATA SCIENTIST';
   ELSE
      SET job_profile = 'MANAGER';
   END IF;
```

```
    RETURN job_profile;
END //

DELIMITER ;

SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP AS EXPERIENCE, GetJobProfile(EXP) AS JOB_PROFILE
FROM emp_record_table
LIMIT 0, 1000;



ALTER TABLE emp_record_table
MODIFY COLUMN FIRST_NAME VARCHAR(255); -- Change the data type as needed

-- Drop the existing index
DROP INDEX idx_first_name ON emp_record_table;

-- Create the new index
CREATE INDEX idx_first_name ON emp_record_table (FIRST_NAME);

EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric';



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



SELECT CONTINENT, COUNTRY, AVG(SALARY) AS AVERAGE_SALARY
FROM emp_record_table
GROUP BY CONTINENT, COUNTRY
```

```
ORDER BY CONTINENT, COUNTRY;
```