```
-- Create the 'employee' database
create database employee;

-- Select the 'employee' database
use employee;

-- View all records from emp_record_table
select* from emp_record_table;

-- Add primary key to EMP_ID column
alter table emp_record_table
add primary key (emp_id(20));


-- Fetch employee ID, name, gender, and department
-- question 3
select
   emp_id,
   first_name,
   last_name,
   gender,
   dept
from emp_record_table;



-- question 4
-- Employees with rating less than 2
select
	emp_id,
   first_name,
   last_name,
   gender,
   dept,
   emp_rating
from emp_record_table
where emp_rating<2;

-- Employees with rating greater than 4
select
	emp_id,
   first_name,
   last_name,
   gender,
   dept,
   emp_rating
from emp_record_table
where emp_rating>4;
```

```
-- Employees with rating between 2 and 4
select
           emp_id,
   first_name,
   last_name,
   gender,
   dept,
   emp_rating
from emp_record_table
where emp_rating between 2 and 4;
```

```
-- Concatenate first and last name of employees in Finance department
-- question 5
select
   first_name,
   last_name,
   concat(first_name,"  ",last_name)
from emp_record_table
where dept="FINANCE";
```

```
-- List employees who are Managers or Presidents
-- question 6
select
   e.emp_id,
   e.first_name,
   e.last_name,
   e.role,
   count(r.emp_id) as num_reportees
from emp_record_table e
join emp_record_table r
   on e.emp_id = r.manager_id
group by
   e.emp_id,
   e.first_name,
   e.last_name,
   e.role;
```

```
-- Combine Finance and Healthcare department employees
-- question 7
select *
from emp_record_table
where dept="FINANCE"
UNION
select *
from emp_record_table
where dept="HEALTHCARE";
```

```
 -- Show max rating per department using window function
-- question 8
select
   emp_id,
   first_name,
   last_name,
   role,
   dept,
   emp_rating,
   max(emp_rating) over (partition by dept) as max_dept_rating
from emp_record_table;
```

```
-- Show min and max salary grouped by role
-- question 9
select role, min(salary),max(salary)
from emp_record_table
group by role;
```

```
-- Assign rank based on ascending experience
-- question 10
select
   emp_id,
   first_name,
   last_name,
   role,
   dept,
   emp_rating,
   rank() over(order by exp asc) as emp_rank
from emp_record_table;
```

```
-- Select high-salary employees by country
-- question 11
select
   emp_id,
   first_name,
   last_name,
   role,
   dept,
   salary,
   country
from emp_record_table
where salary>6000;
```

```
-- Use a subquery to filter employees with >10 years of experience
-- question 12
select
        emp_id,
        first_name,
   last_name,
   role,
   dept,
   exp as experience,
   salary,
   country
from
(select *from emp_record_table) as slry
where exp>10;


-- Create stored procedure to fetch experienced employees
-- question 13
delimiter $$
create procedure get_emp_exp(in experience int)
begin
        select *from emp_record_table e
   where e.exp>experience;
end$$
delimiter ;
call get_emp_exp(3)



-- Create function to return job profile based on experience
-- question 14
delimiter $$
create procedure job_profile(in exp int,out profile varchar(50))
begin
   if exp > 12 and exp <=16 then
                set profile="MANAGER";
   elseif exp > 10 and exp <=12 then
                set profile="LEAD DATA SCIENTIST";
        elseif exp > 5 and exp<=10 then
                set profile="SENIOR DATA SCIENTIST";
   elseif exp >2  and exp<=5 then
                set profile="ASSOCIATE DATA SCIENTIST";
        elseif exp<=2 then
                set profile="JUNIOR DATA SCIENTIST";
        else
                set profile="N/A";

        end if;

end$$
delimiter ;
```

```
call job_profile(15,@job_prof);
select @job_prof as job_profile;
```

```
-- Check initial performance
-- question 15
explain select * from emp_record_table where first_name = 'eric';
create index idx_first_name on employee(first_name);
select * from employee where first_name = 'eric';
explain select * from employee where first_name = 'eric';
```

```
-- Calculate bonus using rating and salary
-- question 16
select
	emp_id,
   first_name,
   role,
   dept,
   emp_rating,
   salary,
   salary*0.05*emp_rating as bonus
from emp_record_table;
```

```
-- Fix syntax: replace 'AND' with ',' inside partition by
-- question 17
select
	emp_id,
   first_name,
   role,
   salary,
   country,
   continent,
   avg(salary) over (partition by continent and country) as average_salary
from emp_record_table;
```