```sql
USE employee_data;
-- 1) Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department
SELECT emp_id, first_name, last_name, gender, dept
FROM emp_record_table ORDER BY emp_id ASC;


-- 2) Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:
-- Less than 2
SELECT emp_id, first_name, last_name, gender, dept, emp_rating
from emp_record_table WHERE emp_rating < 2;


-- Greater than 2
select emp_id, first_name, last_name, gender, dept, emp_rating
from emp_record where emp_rating > 4;


-- Between 2 & 4
select emp_id, first_name, last_name, gender, dept, emp_rating
from emp_record where emp_rating > 2 and emp_rating < 4;


-- 3) Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME
SELECT concat(first_name, ' ', last_name) AS NAME
FROM emp_record_table WHERE dept = 'Finance';


-- 4) Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President)
SELECT manager_id, count(emp_id)
FROM emp_record_table
WHERE manager_id IS NOT NULL
GROUP BY manager_id ORDER BY manager_id ASC;
```

```
-- 5) Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table

SELECT * FROM emp_record_table WHERE dept = 'healthcare'

UNION

SELECT * FROM emp_record_table WHERE dept = 'finance';


-- 6) Write a query to list down employee details such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept

SELECT emp_id, first_name, last_name, role, dept, emp_rating, AVG(emp_rating)

FROM emp_record_table GROUP BY dept;


-- 7) Write a query to calculate the minimum and the maximum salary of the employees in each role

SELECT role, min(emp_rating), max(emp_rating) FROM emp_record_table GROUP BY role;


-- 8) Write a query to assign ranks to each employee based on their experience

SELECT emp_id, first_name, last_name, exp, rank() OVER (ORDER BY exp DESC) AS 'Rank'

FROM emp_record_table;


-- 9) Write a query to create a view that displays employees in various countries whose salary is more than six thousand

CREATE VIEW Test AS SELECT emp_id, first_name, last_name, country, salary

FROM emp_record_table WHERE salary > 6000;


-- 10) Write a nested query to find employees with experience of more than ten years

SELECT * FROM Test;

SELECT * FROM (SELECT * FROM emp_record_table WHERE exp>10) AS tab;


-- 11) Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years

DELIMITER //

CREATE PROCEDURE 3PlusExp()

BEGIN
```

```sql
SELECT *  FROM emp_record_table WHERE exp>3;
END //
delimiter ;
Call 3PlusExp();

-- 12) Write a query using stored functions in the project table to check whether the job profile assigned to each employee in the data science team matches the organization’s set standard
delimiter $$
CREATE FUNCTION check_job_role(exp integer)
RETURNS VARCHAR(40)
DETERMINISTIC
BEGIN
DECLARE chck VARCHAR(40);
if exp < 2 THEN SET chck = "JUNIOR DATA SCIENTIST";
elseif exp >=2 AND exp < 5 THEN SET chck = "ASSOCIATE DATA SCIENTIST";
elseif exp >=5 AND exp < 10 THEN SET chck = "SENIOR DATA SCIENTIST";
elseif exp >= 10 AND exp < 12 THEN SET chck = "LEAD DATA SCIENTIST";
elseif exp >= 12 THEN SET chck = "MANAGER";
end if; RETURN (chck);
END $$
delimiter ;

-- Checking Data Science Team
SELECT emp_id, first_name, last_name, role, check_job_role(exp)
FROM data_science_team WHERE ROLE != check_job_role(exp);

-- 13) Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is ‘Eric’ in the employee table after checking the execution plan
CREATE INDEX FirstName ON emp_record_table (FIRST_NAME(10));

-- 14) Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary * employee rating)
```

```sql
SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, sum((0.05*salary)*emp_rating)
AS comm FROM emp_record_table GROUP BY emp_id ORDER BY emp_id ASC;


-- 15) Write a query to calculate the average salary distribution based on the continent and country
-- Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.
-- Country
select COUNTRY, avg(SALARY) as AVG_SALARY from emp_record
        group by COUNTRY;
-- Continent
select CONTINENT, avg(SALARY) as AVG_SALARY from emp_record
        group by CONTINENT;
```