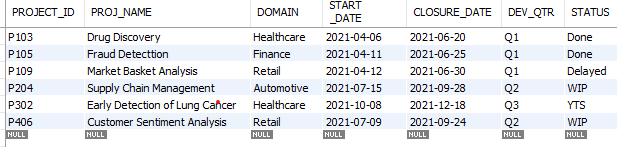
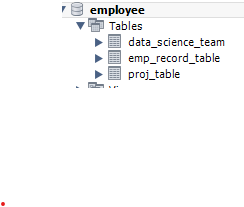
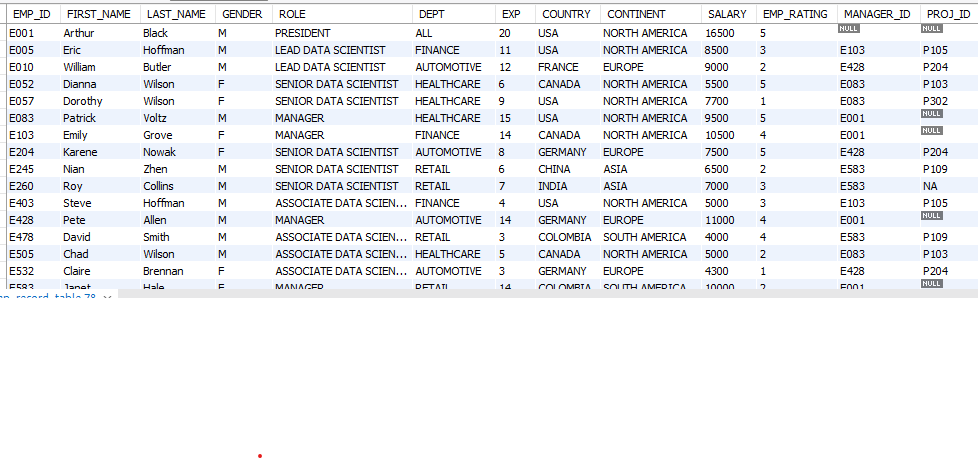
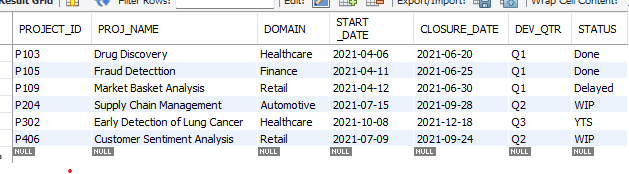
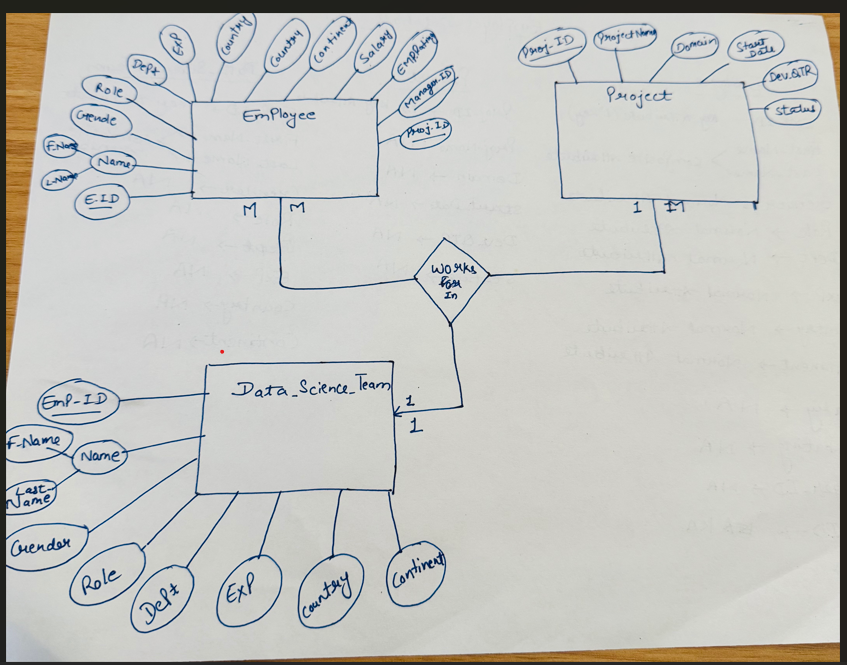
**The task to be performed:**

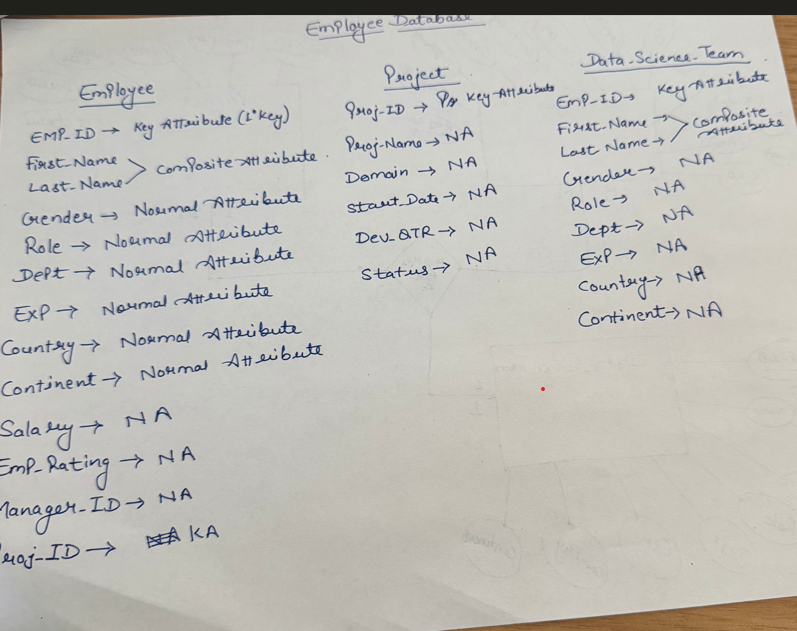
1. Create a database named employee, then import **data\_science\_team.csv** **proj\_table.csv** and **emp\_record\_table.csv** into the **employee**database from the given resources.

->create database employee;

1. Create an ER diagram for the given **employee**database.

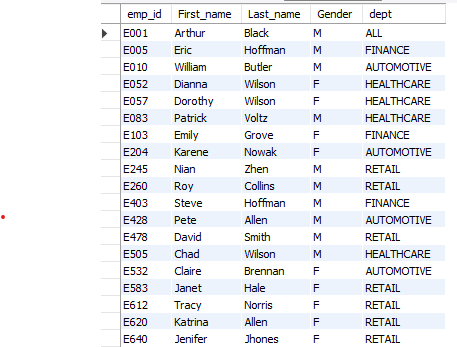




1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

-> select emp\_id,First\_name,Last\_name,Gender,dept

from emp\_record\_table;

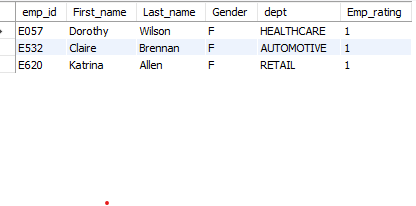


1. Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is:

* less than two
* greater than four
* between two and four
* select emp\_id,First\_name,Last\_name,Gender,dept, Emp\_rating

from emp\_record\_table

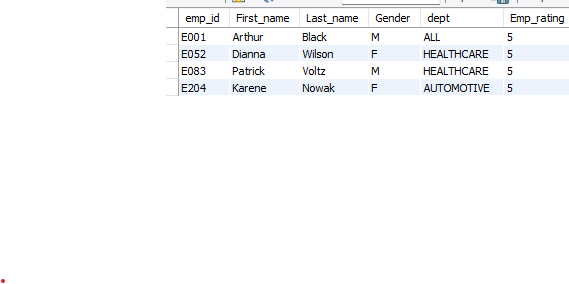
where emp\_rating <2;



* select emp\_id,First\_name,Last\_name,Gender,dept, Emp\_rating

from emp\_record\_table

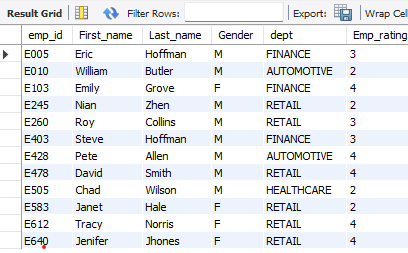
where emp\_rating >4;



* select emp\_id,First\_name,Last\_name, Gender,dept, Emp\_rating

from emp\_record\_table

where emp\_rating between 2 and 4;

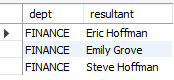


1. Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

* SELECT dept, CONCAT(First\_name, ' ', Last\_Name) AS resultant

FROM emp\_record\_table

WHERE dept = 'finance';



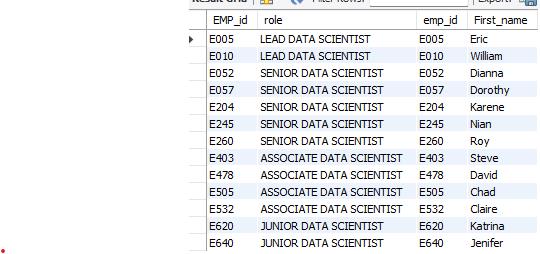
1. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).

-> select Emp\_record\_table.EMP\_id,Emp\_record\_table.role,data\_science\_team.emp\_id,data\_science\_team.First\_name

from emp\_record\_table

inner join data\_science\_team

on emp\_record\_table.emp\_id = data\_science\_team.emp\_id;



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

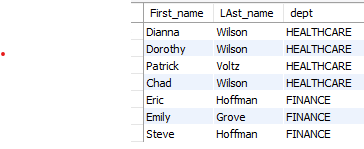
* select First\_name, LAst\_name, dept from emp\_record\_table

where dept = 'healthcare'

union

select First\_name, LAst\_name, dept from emp\_record\_table

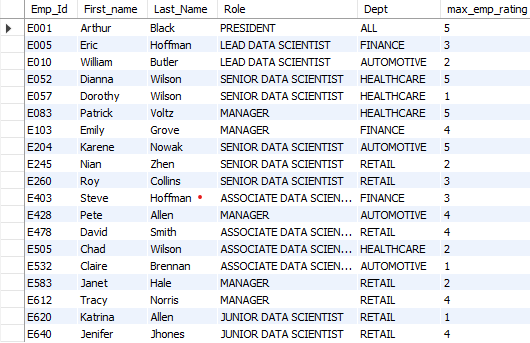
where dept = 'finance';



1. Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department.

select Emp\_Id, First\_name,Last\_Name, Role, Dept, max(EMP\_RATING) as max\_emp\_rating from emp\_record\_table

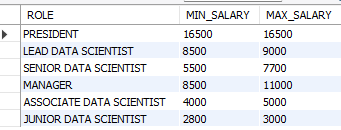
group by Emp\_Id, First\_name,Last\_Name, Role, Dept;



1. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

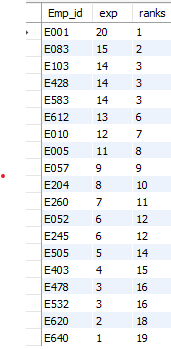
* SELECT ROLE, MIN(SALARY) AS MIN\_SALARY, MAX(SALARY) AS MAX\_SALARY FROM emp\_record\_table

GROUP BY role;



1. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

-> SELECT Emp\_id, exp, RANK() OVER (ORDER BY exp DESC) AS ranks FROM emp\_record\_table;



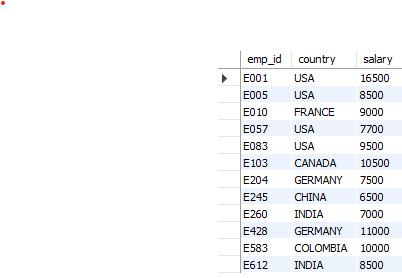
1. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.

-> create view view\_emp as

select emp\_id,country,salary from emp\_record\_table

where salary>6000;

select \* from view\_emp;



1. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

-> select emp\_Id, exp,

case

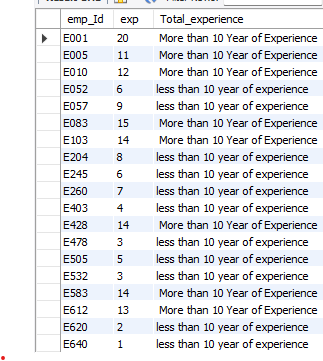
when exp > 10 then ' More than 10 Year of Experience'

else

'less than 10 year of experience'

end as Total\_experience

from emp\_record\_table;



1. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

* DELIMITER $$

USE `employee`$$

CREATE PROCEDURE `get\_experience\_details` ()

BEGIN

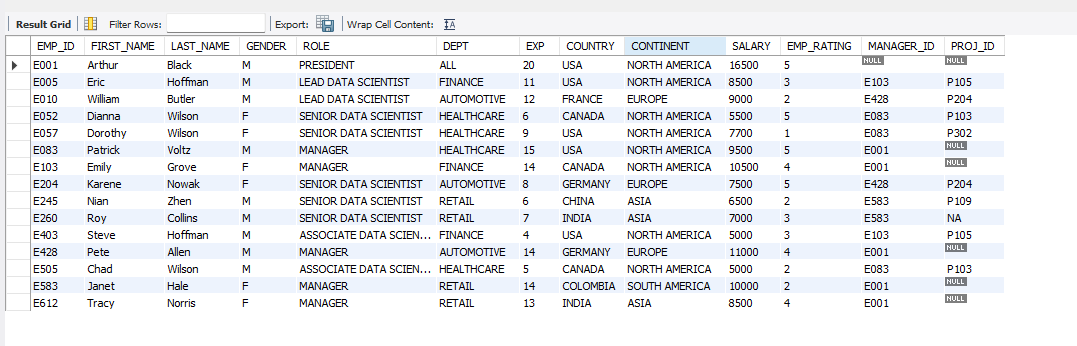
select emp\_id,exp from emp\_record\_table

where exp > 3;

END$$

DELIMITER;

call get\_experience\_details();



1. 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.

The standard being:

For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',

For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',

For an employee with the experience of 5 to 10 years assign 'SENIOR DATA SCIENTIST',

For an employee with the experience of 10 to 12 years assign 'LEAD DATA SCIENTIST',

For an employee with the experience of 12 to 16 years assign 'MANAGER'.

* DELIMITER $$

CREATE FUNCTION checkJobProfile(experience INT, role VARCHAR(100))

RETURNS VARCHAR(100)

returns int

DETERMINISTIC

BEGIN

DECLARE expected\_role VARCHAR(100);

IF experience <= 2 THEN expected role = 'junior data scientist';

select role into expected\_role from employee.data\_science\_team where exp = experience;

IF experience between 2 and 5 THEN expected role = 'Associate data scientist';

select role into expected\_role from employee.data\_science\_team where exp = experience;

IF experience between 5 and 10 THEN expected role = 'Senior data scientist';

select role into expected\_role from employee.data\_science\_team where exp = experience;

IF experience between 10 and 12 THEN expected role = 'Lead data scientist';

select role into expected\_role from employee.data\_science\_team where exp = experience;

IF experience between 12 and 16 THEN expected role = 'Manager';

select role into expected\_role from employee.data\_science\_team where exp = experience;

ELSE

SET expected\_role = 'not exist';

END /

DELIMITER ;

* select checkjobprofile(1);



1. 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 idx\_first\_name ON emp\_record\_table(FIRST\_NAME);

select \* from emp\_record\_table where first\_name = 'Eric';



1. 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).

-> SELECT

EMP\_ID,

FIRST\_NAME,

LAST\_NAME,

SALARY,

EMP\_RATING,

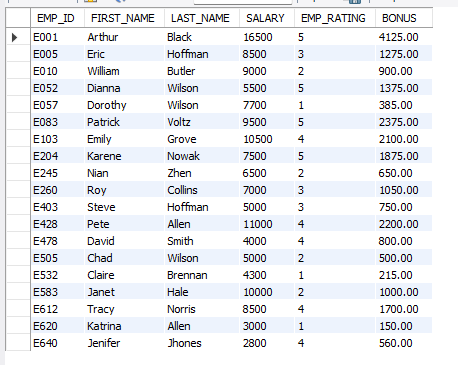
CASE

WHEN EMP\_RATING > 0 THEN Round(0.05 \* SALARY \* EMP\_RATING, 2)

ELSE 0

END AS BONUS

FROM emp\_record\_table;



1. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.

-> SELECT

CONTINENT,

COUNTRY,

CASE

when COUNTRY = 'USA' AND CONTINENT = 'North America' then avg(salary)

WHEN country = 'CANADA' and Continent = 'North America' tHEN avg(salary)

WHEN country = 'france' and Continent = 'Europe' tHEN avg(salary)

WHEN country = 'Germany' and Continent = 'Europe' then avg(salary)

WHEN country = 'China' and Continent = 'Asia' then avg(salary)

WHEN country = 'India' and Continent = 'Asia' then avg(salary)

WHEN country = 'colombia' and Continent = 'South America' then avg(salary)

ELSE NULL

END AS AVG\_SALARY

FROM emp\_record\_table

group by continent, country

