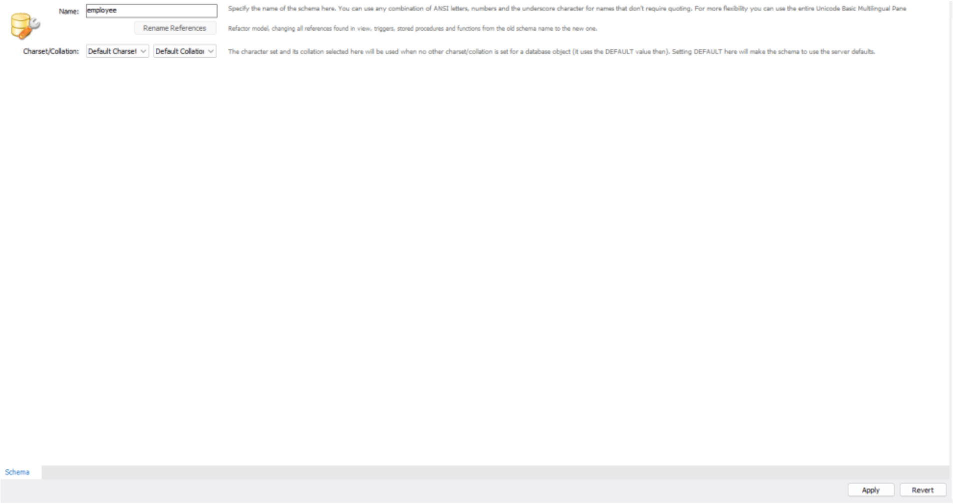
NAME: Hamad Saad Alyemni

CLASS #: 8

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.



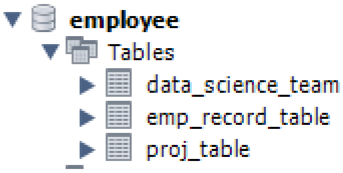
Name: employee

Specify the name of the schema here. You can use any combination of ANSI letters, numbers and the underscore character for names that don't require quoting. For more flexibility you can use the entire Unicode Basic Multilingual Pane

Refactor model, changing all references found in view, triggers, stored procedures and functions from the old schema name to the new one.

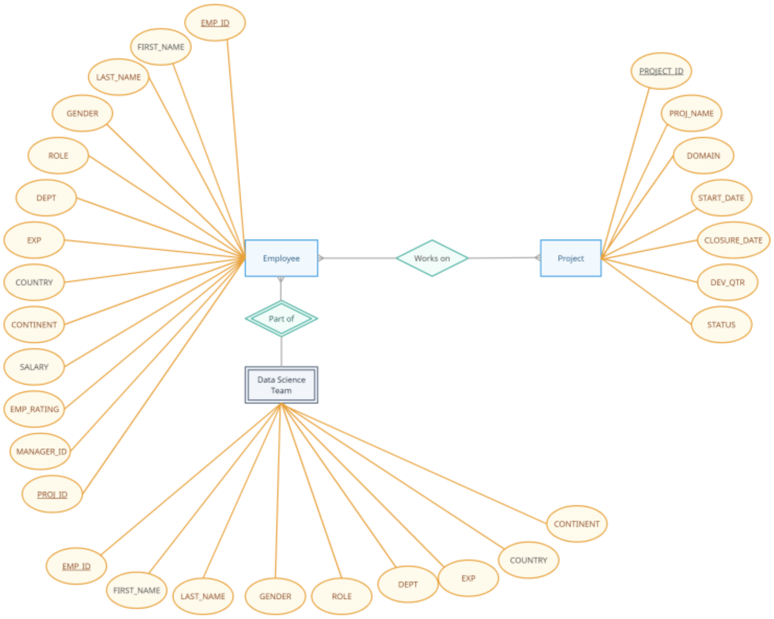
Charset/Collation: Default Charset Default Collation

The character set and its collation selected here will be used when no other charset/collation is set for a database object (it uses the DEFAULT value then). Setting DEFAULT here will make the schema to use the server defaults.



employee
Tables
data_science_team
emp_record_table
proj_table

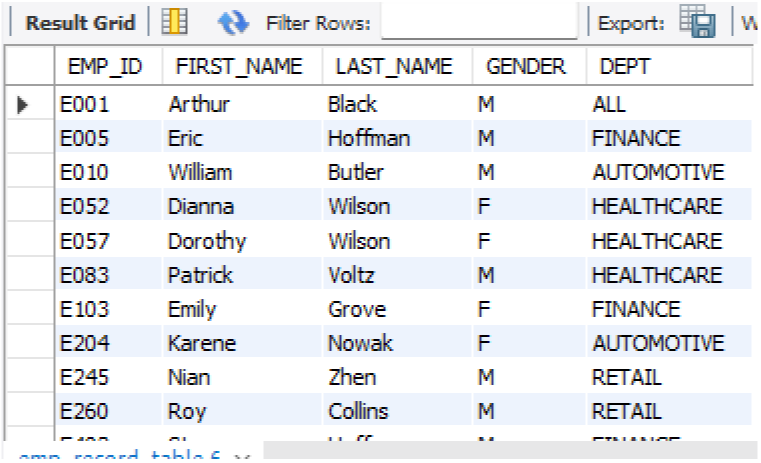
2. Create an ER diagram for the given employee database.



3. 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;

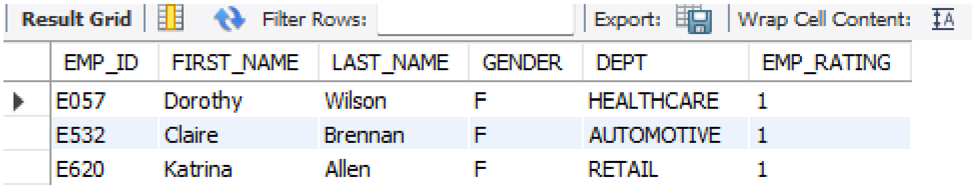


EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E001	Arthur	Black	M	ALL
E005	Eric	Hoffman	M	FINANCE
E010	William	Butler	M	AUTOMOTIVE
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E103	Emily	Grove	F	FINANCE
E204	Karene	Nowak	F	AUTOMOTIVE
E245	Nian	Zhen	M	RETAIL
E260	Roy	Collins	M	RETAIL

4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

less than two

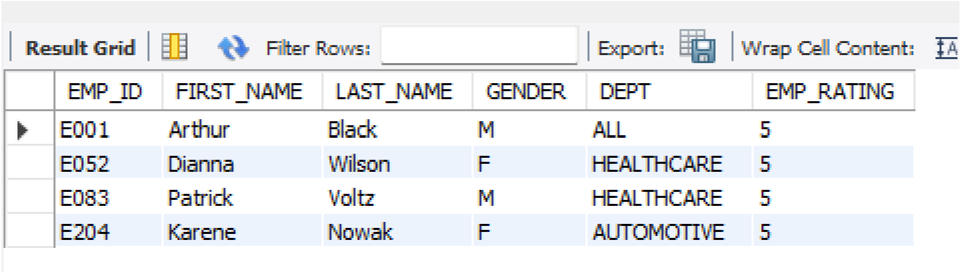
```
SELECT EMP_ID ,FIRST_NAME ,LAST_NAME ,GENDER ,DEPT ,EMP_RATING

FROM employee.emp_record_table

WHERE EMP_RATING<2;
```



EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E057	Dorothy	Wilson	F	HEALTHCARE	1
E532	Claire	Brennan	F	AUTOMOTIVE	1
E620	Katrina	Allen	F	RETAIL	1

greater than four

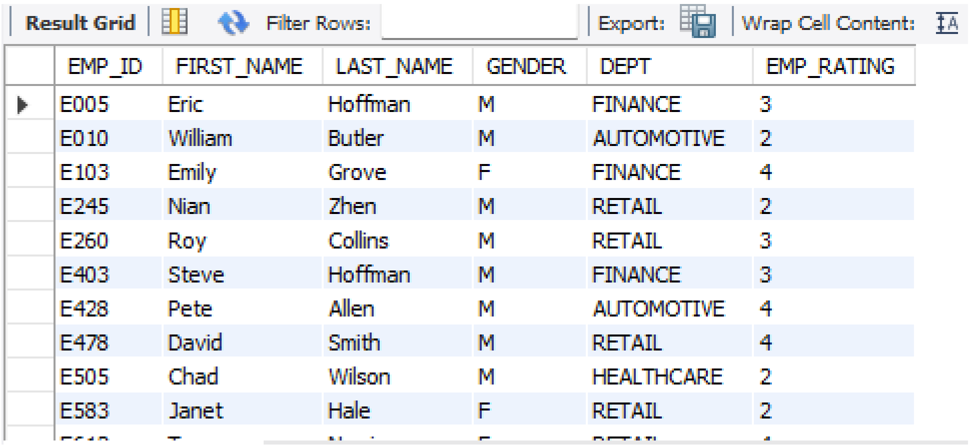
```
SELECT EMP_ID ,FIRST_NAME ,LAST_NAME ,GENDER ,DEPT ,EMP_RATING

FROM employee.emp_record_table

WHERE EMP_RATING>4;
```



EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E001	Arthur	Black	M	ALL	5
E052	Dianna	Wilson	F	HEALTHCARE	5
E083	Patrick	Voltz	M	HEALTHCARE	5
E204	Karene	Nowak	F	AUTOMOTIVE	5

between two and four

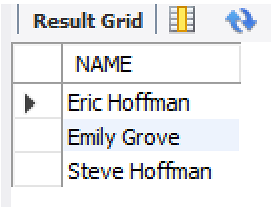
```
SELECT EMP_ID ,FIRST_NAME ,LAST_NAME ,GENDER ,DEPT ,EMP_RATING

FROM employee.emp_record_table

WHERE EMP_RATING BETWEEN 2 AND 4;
```



EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E005	Eric	Hoffman	M	FINANCE	3
E010	William	Butler	M	AUTOMOTIVE	2
E103	Emily	Grove	F	FINANCE	4
E245	Nian	Zhen	M	RETAIL	2
E260	Roy	Collins	M	RETAIL	3
E403	Steve	Hoffman	M	FINANCE	3
E428	Pete	Allen	M	AUTOMOTIVE	4
E478	David	Smith	M	RETAIL	4
E505	Chad	Wilson	M	HEALTHCARE	2
E583	Janet	Hale	F	RETAIL	2

5. 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.

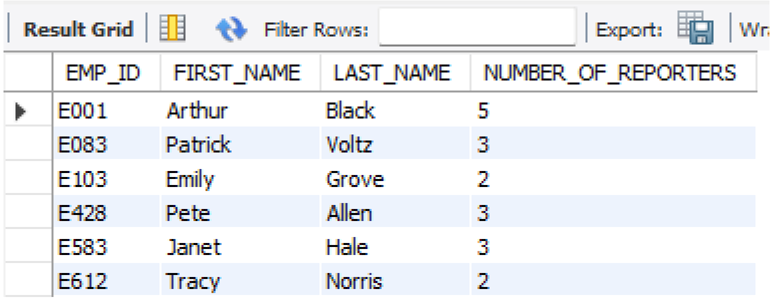
```
SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME

FROM emp_record_table

WHERE DEPT = 'Finance';
```



NAME
Eric Hoffman
Emily Grove
Steve Hoffman

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

```
SELECT  E.EMP_ID, E.FIRST_NAME, E.LAST_NAME, COUNT(DISTINCT R.EMP_ID) AS
NUMBER_OF_REPORTERS

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

HAVING COUNT(DISTINCT R.EMP_ID) > 0;
```

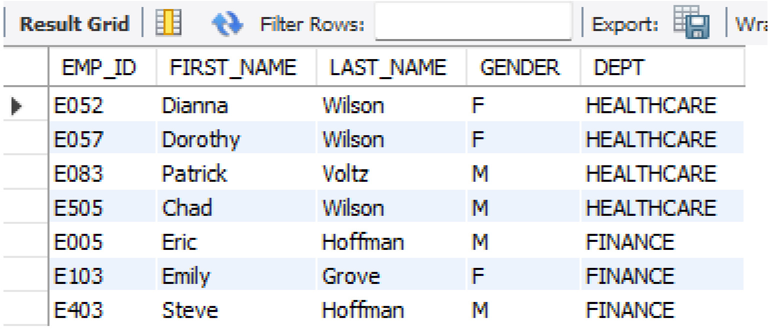


EMP_ID	FIRST_NAME	LAST_NAME	NUMBER_OF_REPORTERS
E001	Arthur	Black	5
E083	Patrick	Voltz	3
E103	Emily	Grove	2
E428	Pete	Allen	3
E583	Janet	Hale	3
E612	Tracy	Norris	2

7. 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 EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT

FROM emp_record_table

WHERE DEPT = 'Healthcare'

UNION
```

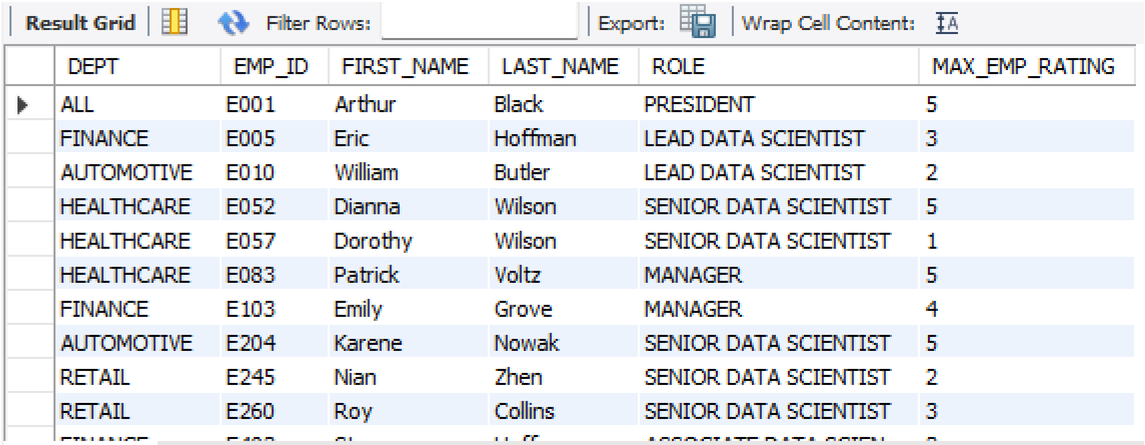
```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT

FROM emp_record_table

WHERE DEPT = 'Finance';
```



EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
E052	Dianna	Wilson	F	HEALTHCARE
E057	Dorothy	Wilson	F	HEALTHCARE
E083	Patrick	Voltz	M	HEALTHCARE
E505	Chad	Wilson	M	HEALTHCARE
E005	Eric	Hoffman	M	FINANCE
E103	Emily	Grove	F	FINANCE
E403	Steve	Hoffman	M	FINANCE

8. 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.

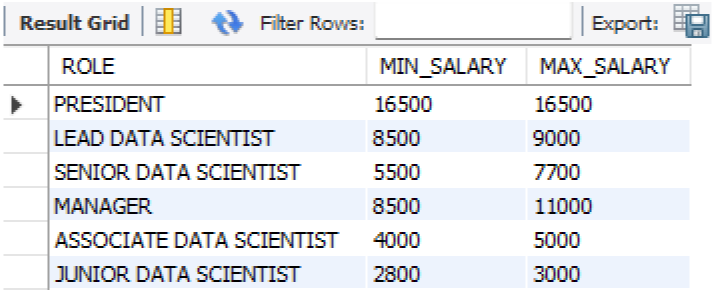
```
SELECT DEPT, EMP_ID, FIRST_NAME, LAST_NAME, ROLE, MAX(EMP_RATING) AS MAX_EMP_RATING

FROM emp_record_table

GROUP BY DEPT, EMP_ID, FIRST_NAME, LAST_NAME, ROLE;
```



DEPT	EMP_ID	FIRST_NAME	LAST_NAME	ROLE	MAX_EMP_RATING
ALL	E001	Arthur	Black	PRESIDENT	5
FINANCE	E005	Eric	Hoffman	LEAD DATA SCIENTIST	3
AUTOMOTIVE	E010	William	Butler	LEAD DATA SCIENTIST	2
HEALTHCARE	E052	Dianna	Wilson	SENIOR DATA SCIENTIST	5
HEALTHCARE	E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	1
HEALTHCARE	E083	Patrick	Voltz	MANAGER	5
FINANCE	E103	Emily	Grove	MANAGER	4
AUTOMOTIVE	E204	Karene	Nowak	SENIOR DATA SCIENTIST	5
RETAIL	E245	Nian	Zhen	SENIOR DATA SCIENTIST	2
RETAIL	E260	Roy	Collins	SENIOR DATA SCIENTIST	3
FINANCE	E403	St	Hoff	ASSOCIATE DATA SCIEN	2

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

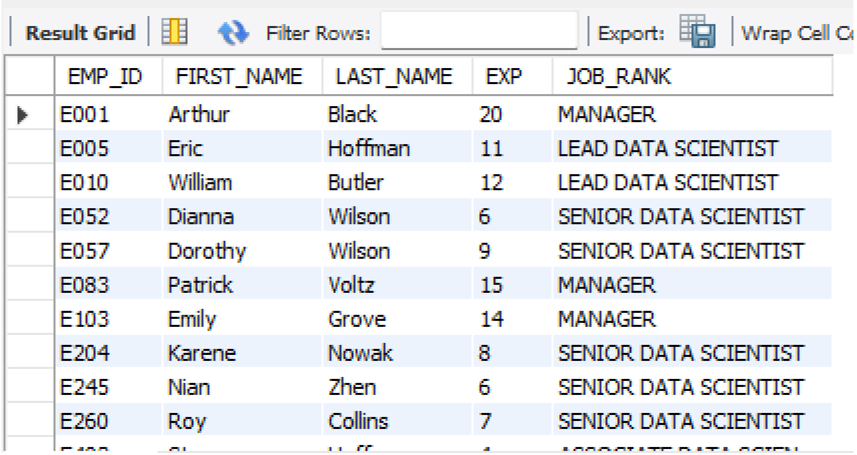
```
SELECT ROLE, MIN(SALARY) AS MINIMUM_SALARY,  MAX(SALARY) AS MAXIMUM_SALARY

FROM emp_record_table

GROUP BY ROLE;
```



ROLE	MIN_SALARY	MAX_SALARY
PRESIDENT	16500	16500
LEAD DATA SCIENTIST	8500	9000
SENIOR DATA SCIENTIST	5500	7700
MANAGER	8500	11000
ASSOCIATE DATA SCIENTIST	4000	5000
JUNIOR DATA SCIENTIST	2800	3000

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

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,
 CASE
  WHEN EXP <= 2 THEN 'JUNIOR DATA SCIENTIST'
  WHEN EXP <= 5 THEN 'ASSOCIATE DATA SCIENTIST'
  WHEN EXP <= 10 THEN 'SENIOR DATA SCIENTIST'
  WHEN EXP <= 12 THEN 'LEAD DATA SCIENTIST'
  ELSE 'MANAGER'
 END AS JOB_RANK
FROM emp_record_table;
```



EMP_ID	FIRST_NAME	LAST_NAME	EXP	JOB_RANK
E001	Arthur	Black	20	MANAGER
E005	Eric	Hoffman	11	LEAD DATA SCIENTIST
E010	William	Butler	12	LEAD DATA SCIENTIST
E052	Dianna	Wilson	6	SENIOR DATA SCIENTIST
E057	Dorothy	Wilson	9	SENIOR DATA SCIENTIST
E083	Patrick	Voltz	15	MANAGER
E103	Emily	Grove	14	MANAGER
E204	Karene	Nowak	8	SENIOR DATA SCIENTIST
E245	Nian	Zhen	6	SENIOR DATA SCIENTIST
E260	Roy	Collins	7	SENIOR DATA SCIENTIST

11. 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.

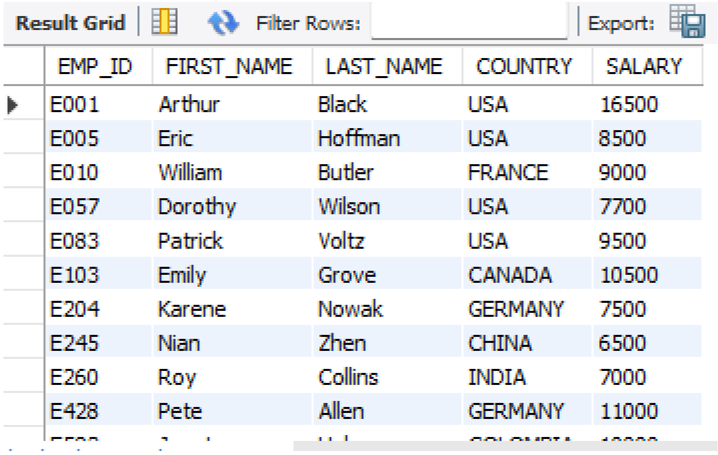
```
CREATE VIEW HighSalaryEmployees AS
```

SELECT EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY

FROM emp_record_table

WHERE SALARY > 6000;

SELECT * FROM highsalaryemployees;



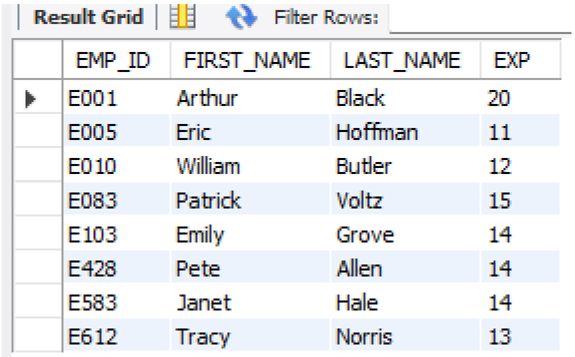
EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
E001	Arthur	Black	USA	16500
E005	Eric	Hoffman	USA	8500
E010	William	Butler	FRANCE	9000
E057	Dorothy	Wilson	USA	7700
E083	Patrick	Voltz	USA	9500
E103	Emily	Grove	CANADA	10500
E204	Karene	Nowak	GERMANY	7500
E245	Nian	Zhen	CHINA	6500
E260	Roy	Collins	INDIA	7000
E428	Pete	Allen	GERMANY	11000

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

SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP

FROM emp_record_table

WHERE EXP > (SELECT MAX(EXP) FROM emp_record_table WHERE EXP <= 10);

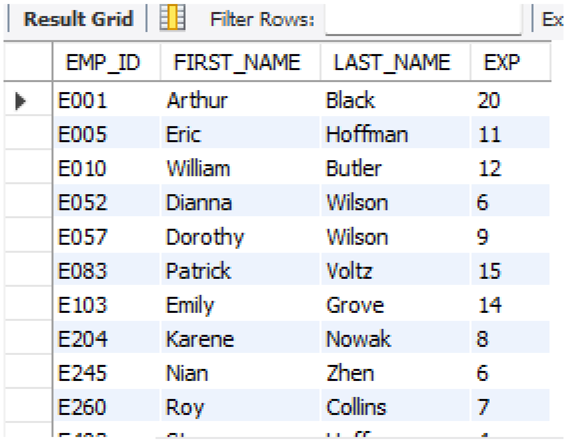


EMP_ID	FIRST_NAME	LAST_NAME	EXP
E001	Arthur	Black	20
E005	Eric	Hoffman	11
E010	William	Butler	12
E083	Patrick	Voltz	15
E103	Emily	Grove	14
E428	Pete	Allen	14
E583	Janet	Hale	14
E612	Tracy	Norris	13

13. 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 //

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



EMP_ID	FIRST_NAME	LAST_NAME	EXP
E001	Arthur	Black	20
E005	Eric	Hoffman	11
E010	William	Butler	12
E052	Dianna	Wilson	6
E057	Dorothy	Wilson	9
E083	Patrick	Voltz	15
E103	Emily	Grove	14
E204	Karene	Nowak	8
E245	Nian	Zhen	6
E260	Roy	Collins	7

14. 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',

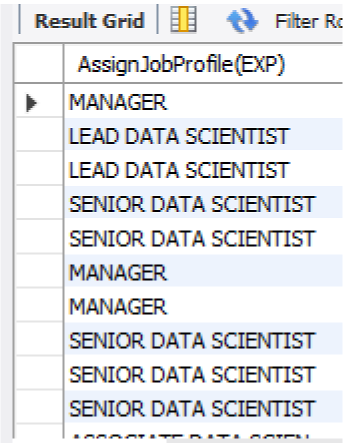
For an employee with the experience of 10 to 12 years assign 'LEAD DATA SCIENTIST',

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

```
 DELIMITER //
CREATE FUNCTION AssignJobProfile(EXP INT)
RETURNS VARCHAR(50) DETERMINISTIC
BEGIN
   DECLARE JobProfile VARCHAR(50);

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

   RETURN JobProfile;
END //
DELIMITER ;
```

SELECT AssignJobProfile(EXP)

FROM emp_record_table;

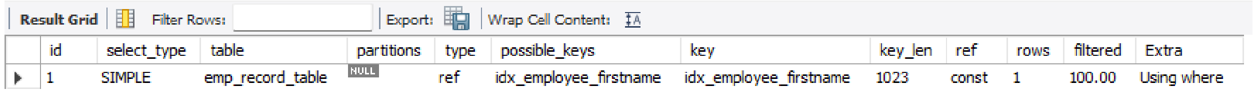


AssignJobProfile(EXP)
MANAGER
LEAD DATA SCIENTIST
LEAD DATA SCIENTIST
SENIOR DATA SCIENTIST
SENIOR DATA SCIENTIST
MANAGER
MANAGER
SENIOR DATA SCIENTIST
SENIOR DATA SCIENTIST
SENIOR DATA SCIENTIST
ASSOCIATE DATA SCIEN

15. 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_employee_firstname ON emp_record_table (FIRST_NAME(255));

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

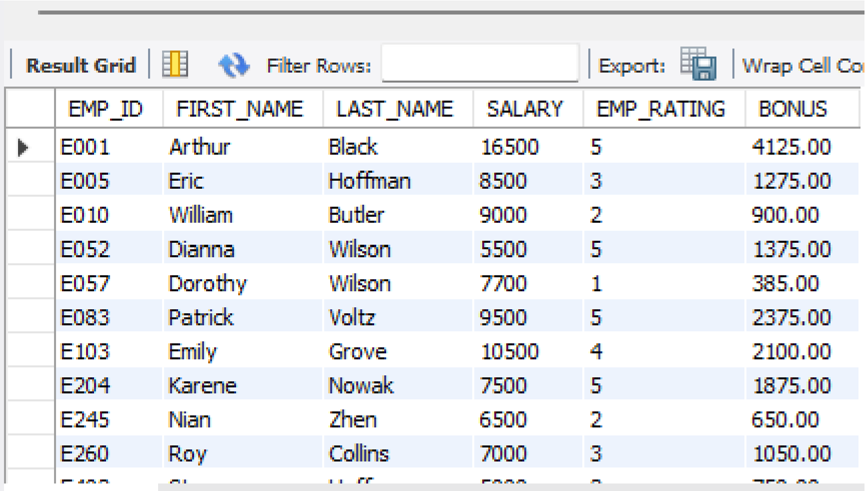


id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	emp_record_table	NULL	ref	idx_employee_firstname	idx_employee_firstname	1023	const	1	100.00	Using where

16. 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, (SALARY * 0.05 * EMP_RATING) AS BONUS

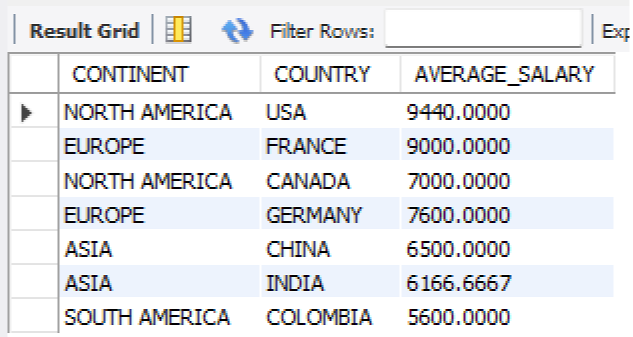
FROM emp_record_table;



EMP_ID	FIRST_NAME	LAST_NAME	SALARY	EMP_RATING	BONUS
E001	Arthur	Black	16500	5	4125.00
E005	Eric	Hoffman	8500	3	1275.00
E010	William	Butler	9000	2	900.00
E052	Dianna	Wilson	5500	5	1375.00
E057	Dorothy	Wilson	7700	1	385.00
E083	Patrick	Voltz	9500	5	2375.00
E103	Emily	Grove	10500	4	2100.00
E204	Karene	Nowak	7500	5	1875.00
E245	Nian	Zhen	6500	2	650.00
E260	Roy	Collins	7000	3	1050.00

17. 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, AVG(SALARY) AS AVERAGE_SALARY

FROM emp_record_table

GROUP BY CONTINENT, COUNTRY;
```



CONTINENT	COUNTRY	AVERAGE_SALARY
NORTH AMERICA	USA	9440.0000
EUROPE	FRANCE	9000.0000
NORTH AMERICA	CANADA	7000.0000
EUROPE	GERMANY	7600.0000
ASIA	CHINA	6500.0000
ASIA	INDIA	6166.6667
SOUTH AMERICA	COLOMBIA	5600.0000