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.

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.

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
- greater than four
- between two and four

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.

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

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.

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.

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.

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

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.

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

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.

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'.

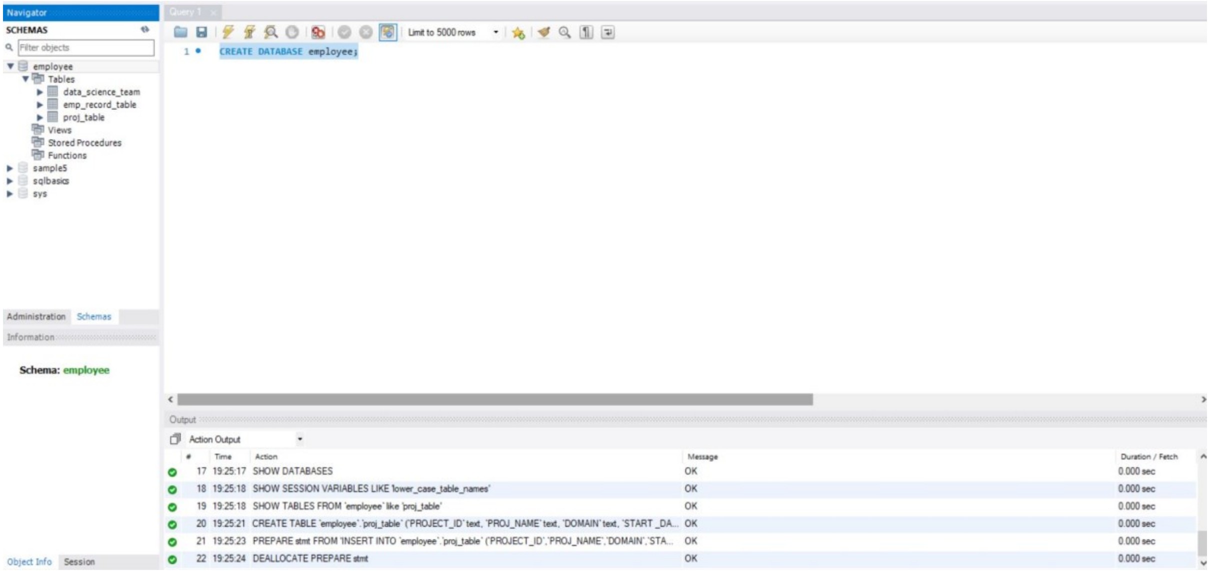
15. Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is ‘Eric’ in the employee table after checking the execution plan.

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).

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

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.

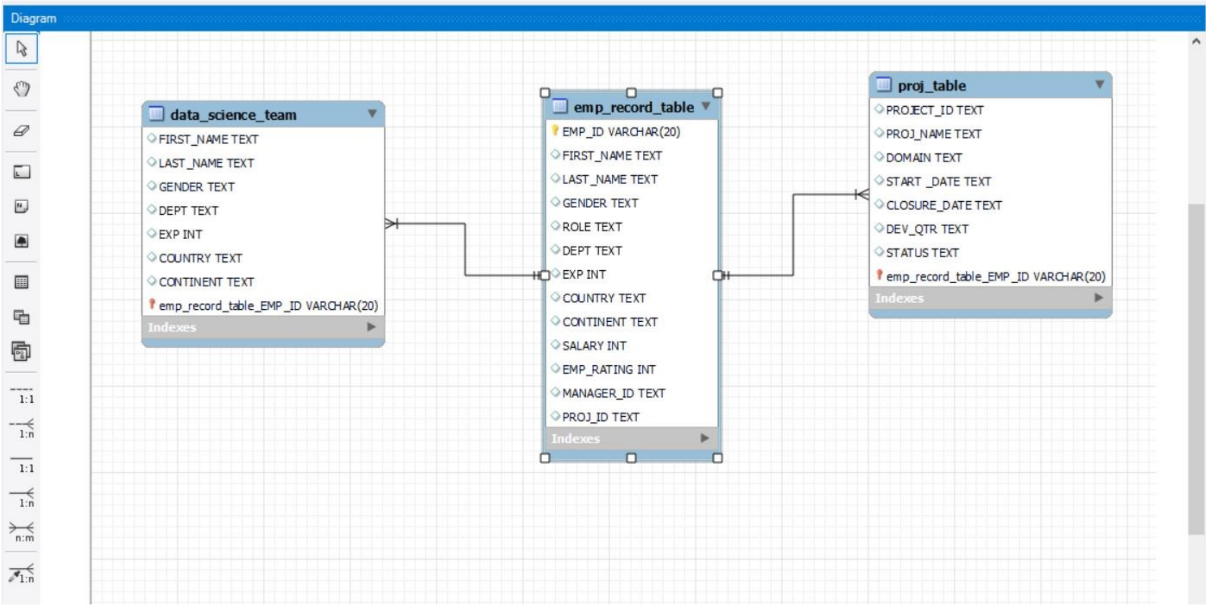
1A. CREATE DATABASE employee;



```
CREATE DATABASE employee;
```

```
#   Time      Action                                                                                          Message   Duration / Fetch
17  19:25:17  SHOW DATABASES                                                                                  OK        0.000 sec
18  19:25:18  SHOW SESSION VARIABLES LIKE 'lower_case_table_names'                                            OK        0.000 sec
19  19:25:18  SHOW TABLES FROM `employee` like 'proj_table'                                                  OK        0.000 sec
20  19:25:21  CREATE TABLE `employee`.`proj_table` (`PROJECT_ID` text, `PROJ_NAME` text, `DOMAIN` text, `START _DA...  OK   0.000 sec
21  19:25:23  PREPARE stmt FROM 'INSERT INTO `employee`.`proj_table` (`PROJECT_ID`,`PROJ_NAME`,`DOMAIN`,`STA...       OK   0.000 sec
22  19:25:24  DEALLOCATE PREPARE stmt                                                                         OK        0.000 sec
```

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

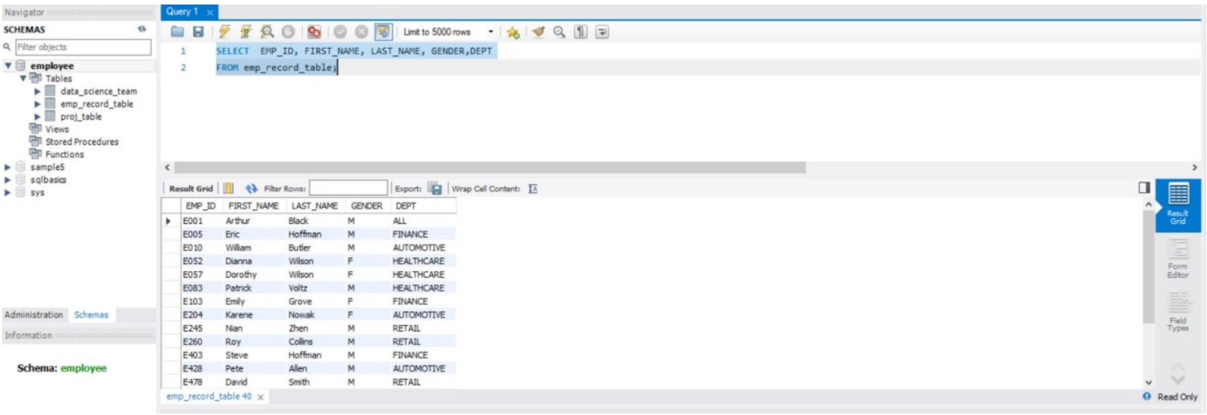


3.Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

3A.

SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER,DEPT

FROM emp_record_table;



```
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
E403	Steve	Hoffman	M	FINANCE
E428	Pete	Allen	M	AUTOMOTIVE
E478	David	Smith	M	RETAIL

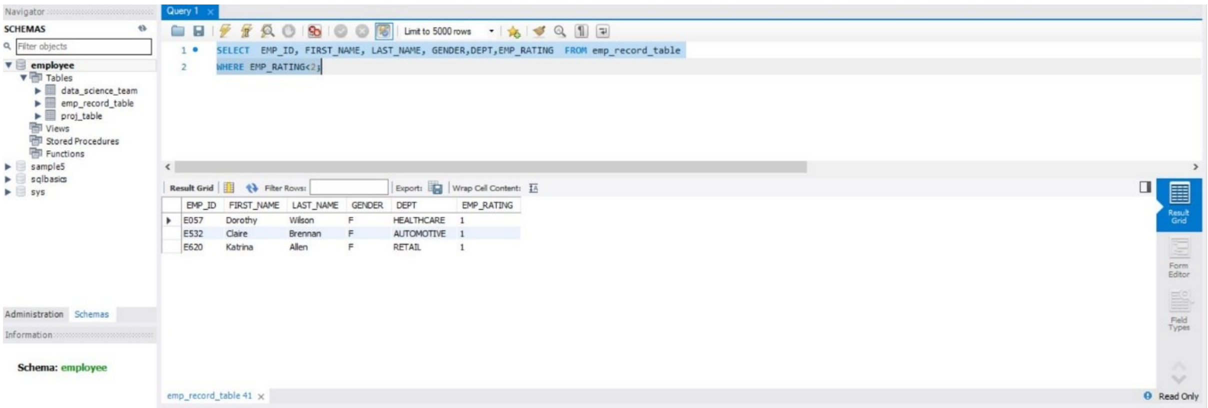
4.Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

- less than two
- greater than four
- between two and four

4A.   less than two

```
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<2;
```

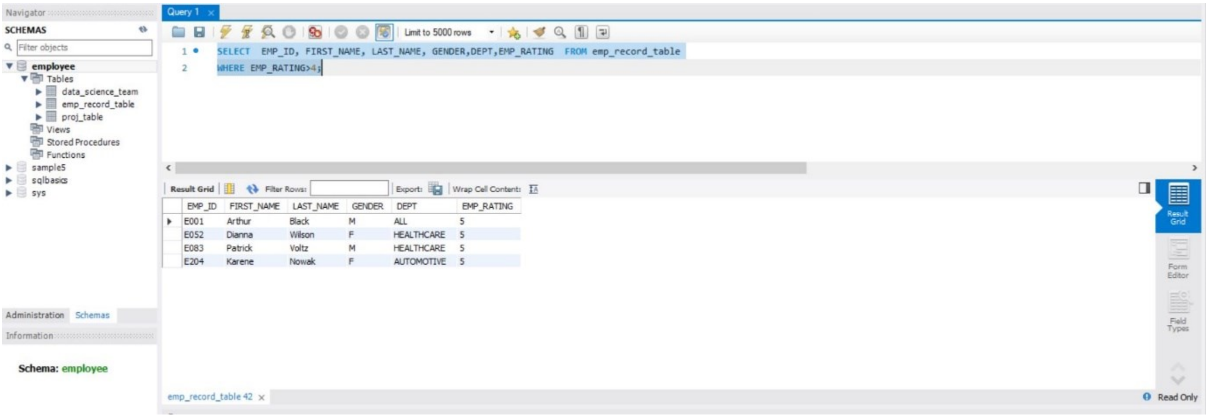
EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E057	Dorothy	Wilson	F	HEALTHCARE	1
E532	Claire	Brennan	F	AUTOMOTIVE	1
E620	Katrina	Allen	F	RETAIL	1

4B.

greater than four

```
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>4;
```

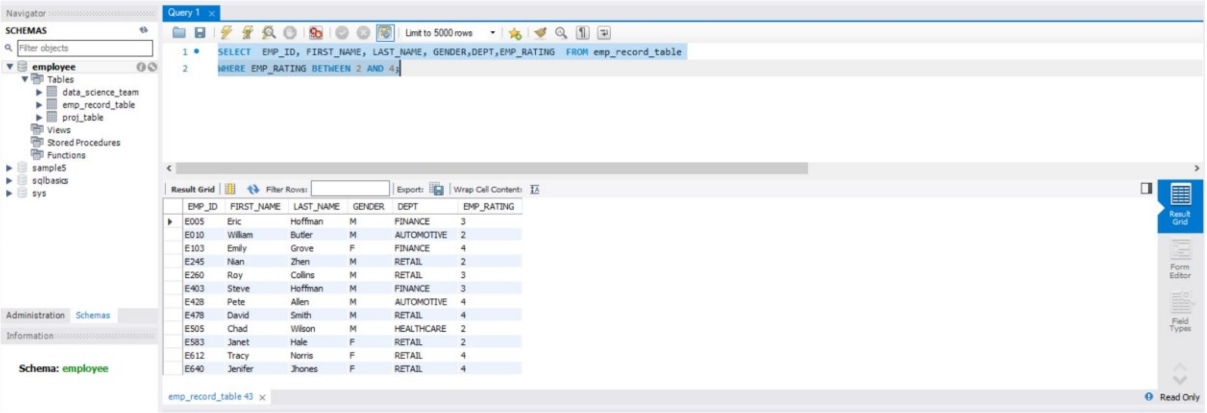
EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E001	Arthur	Black	M	ALL	5
E052	Dianna	Wilson	F	HEALTHCARE	5
E083	Patrick	Voltz	M	HEALTHCARE	5
E204	Karene	Nowak	F	AUTOMOTIVE	5

4C.

between two and four

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

WHERE EMP_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;
```

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
E612	Tracy	Norris	F	RETAIL	4
E640	Jenifer	Jhones	F	RETAIL	4

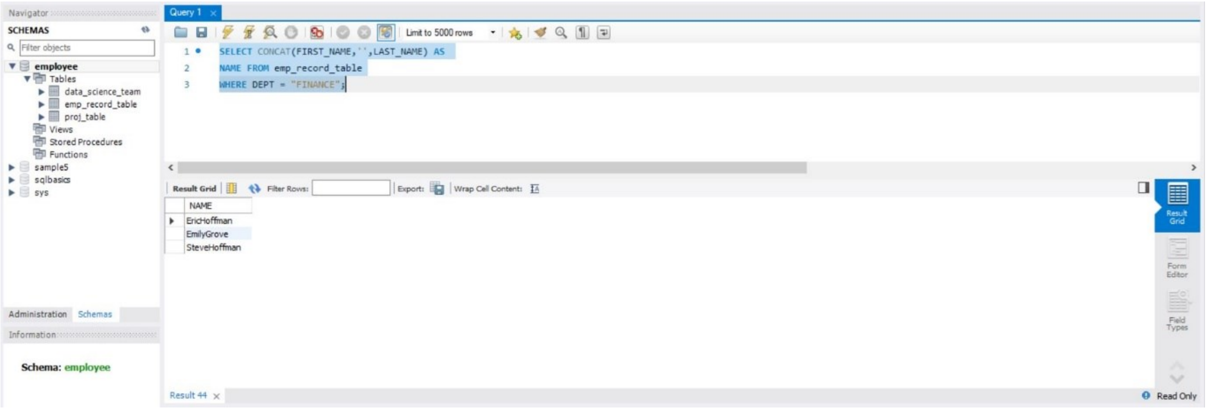
5.Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

5A.

SELECT CONCAT(FIRST_NAME,'',LAST_NAME) AS

NAME FROM emp_record_table

WHERE DEPT = "FINANCE";



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

NAME
EricHoffman
EmilyGrove
SteveHoffman

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

6A.

SELECT m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,

m.EXP,COUNT(e.EMP_ID) as "EMP_COUNT"

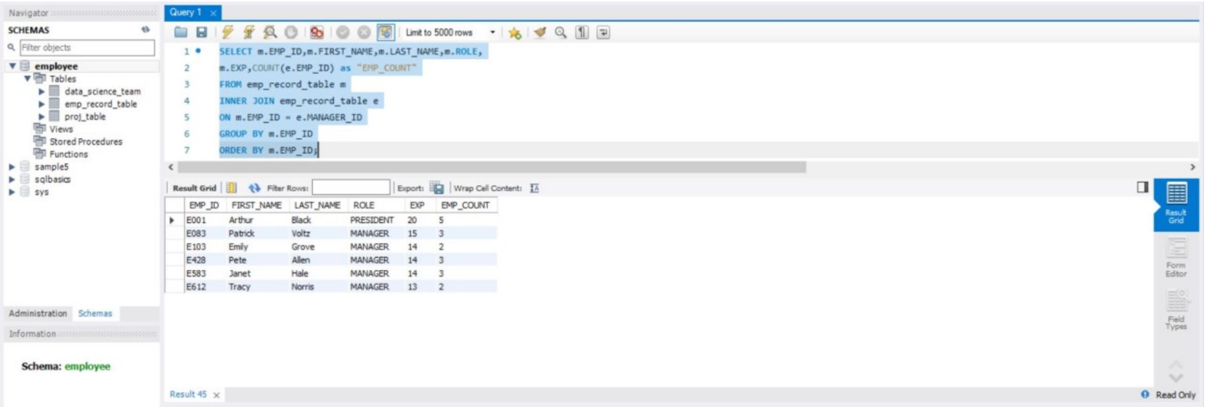
FROM emp_record_table m

INNER JOIN emp_record_table e

ON m.EMP_ID = e.MANAGER_ID

GROUP BY m.EMP_ID

ORDER BY m.EMP_ID;



```
SELECT m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,
m.EXP,COUNT(e.EMP_ID) as "EMP_COUNT"
FROM emp_record_table m
INNER JOIN emp_record_table e
ON m.EMP_ID = e.MANAGER_ID
GROUP BY m.EMP_ID
ORDER BY m.EMP_ID;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	EXP	EMP_COUNT
E001	Arthur	Black	PRESIDENT	20	5
E083	Patrick	Voltz	MANAGER	15	3
E103	Emily	Grove	MANAGER	14	2
E428	Pete	Allen	MANAGER	14	3
E583	Janet	Hale	MANAGER	14	3
E612	Tracy	Norris	MANAGER	13	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.

7A.

SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT FROM emp_record_table

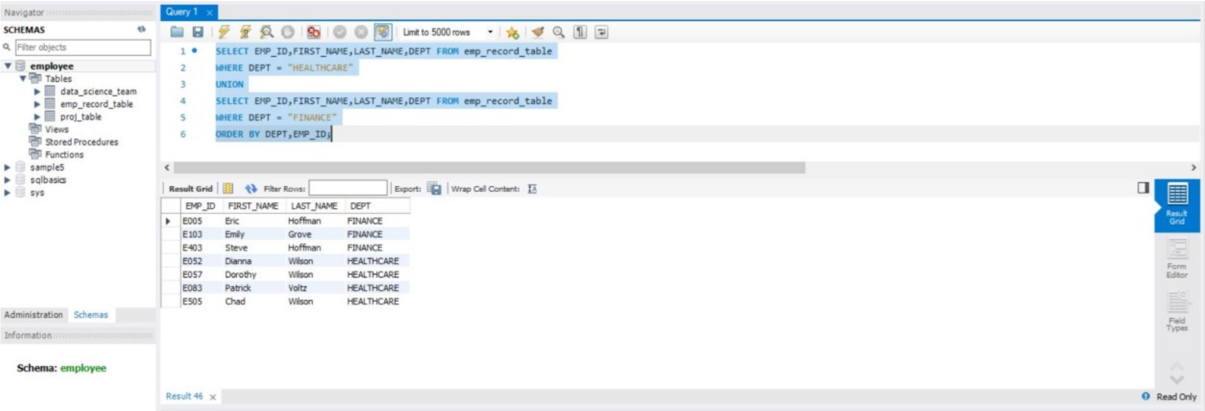
WHERE DEPT = "HEALTHCARE"

UNION

SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT FROM emp_record_table

WHERE DEPT = "FINANCE"

ORDER BY DEPT,EMP_ID;



```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT FROM emp_record_table
WHERE DEPT = "HEALTHCARE"
UNION
SELECT EMP_ID,FIRST_NAME,LAST_NAME,DEPT FROM emp_record_table
WHERE DEPT = "FINANCE"
ORDER BY DEPT,EMP_ID;
```

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

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.

8A.

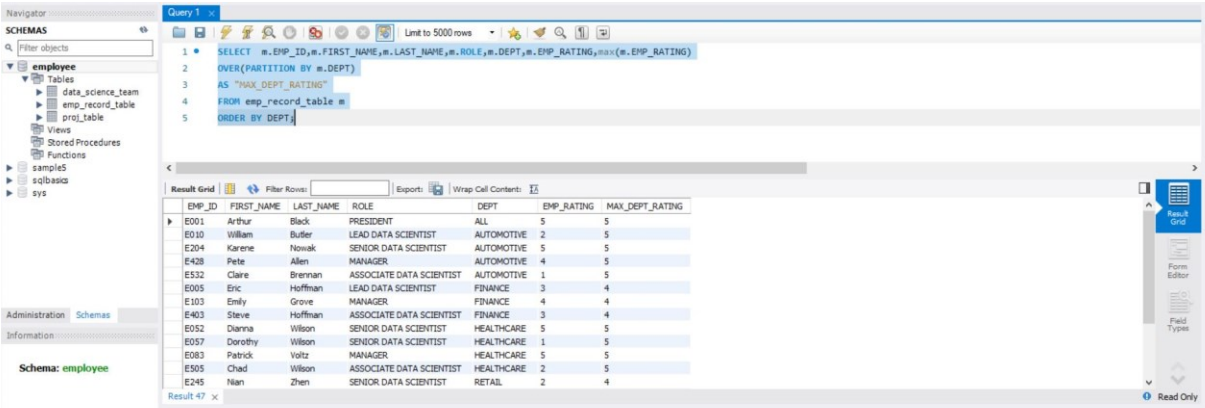
SELECT m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,m.DEPT,m.EMP_RATING,max(m.EMP_RATING)

OVER(PARTITION BY m.DEPT)

AS "MAX_DEPT_RATING"

FROM emp_record_table m

ORDER BY DEPT;



```
SELECT  m.EMP_ID,m.FIRST_NAME,m.LAST_NAME,m.ROLE,m.DEPT,m.EMP_RATING,max(m.EMP_RATING)
OVER(PARTITION BY m.DEPT)
AS "MAX_DEPT_RATING"
FROM emp_record_table m
ORDER BY DEPT;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	MAX_DEPT_RATING
E001	Arthur	Black	PRESIDENT	ALL	5	5
E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
E103	Emily	Grove	MANAGER	FINANCE	4	4
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
E245	Nian	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4

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.

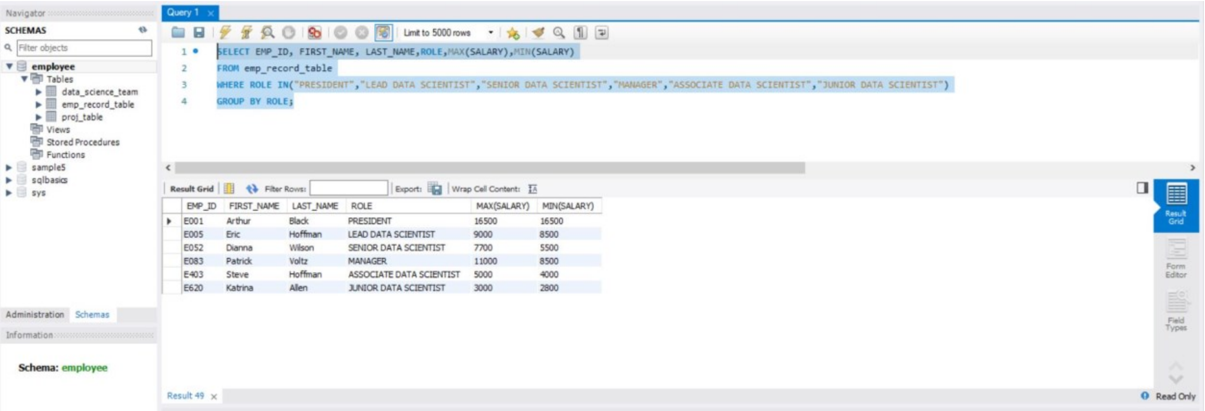
9A.

SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, MAX(SALARY), MIN(SALARY)

FROM emp_record_table

WHERE ROLE IN("PRESIDENT","LEAD DATA SCIENTIST","SENIOR DATA SCIENTIST","MANAGER","ASSOCIATE DATA SCIENTIST","JUNIOR DATA SCIENTIST")

GROUP BY ROLE;



```
SELECT EMP_ID, FIRST_NAME, LAST_NAME,ROLE,MAX(SALARY),MIN(SALARY)
FROM emp_record_table
WHERE ROLE IN("PRESIDENT","LEAD DATA SCIENTIST","SENIOR DATA SCIENTIST","MANAGER","ASSOCIATE DATA SCIENTIST","JUNIOR DATA SCIENTIST")
GROUP BY ROLE;
```

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	MAX(SALARY)	MIN(SALARY)
E001	Arthur	Black	PRESIDENT	16500	16500
E005	Eric	Hoffman	LEAD DATA SCIENTIST	9000	8500
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	7700	5500
E083	Patrick	Voltz	MANAGER	11000	8500
E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	5000	4000
E620	Katrina	Allen	JUNIOR DATA SCIENTIST	3000	2800

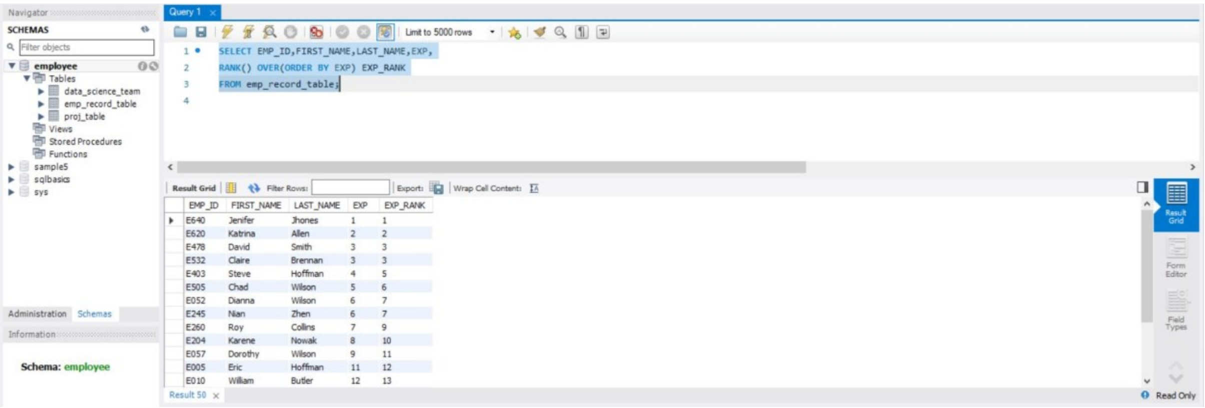
10.Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

10A.

SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP,

RANK() OVER(ORDER BY EXP) EXP_RANK

FROM emp_record_table;



```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP,
RANK() OVER(ORDER BY EXP) EXP_RANK
FROM emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	EXP	EXP_RANK
E640	Jenifer	Jhones	1	1
E620	Katrina	Allen	2	2
E478	David	Smith	3	3
E532	Claire	Brennan	3	3
E403	Steve	Hoffman	4	5
E505	Chad	Wilson	5	6
E052	Dianna	Wilson	6	7
E245	Nian	Zhen	6	7
E260	Roy	Collins	7	9
E204	Karene	Nowak	8	10
E057	Dorothy	Wilson	9	11
E005	Eric	Hoffman	11	12
E010	William	Butler	12	13

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.

11A.

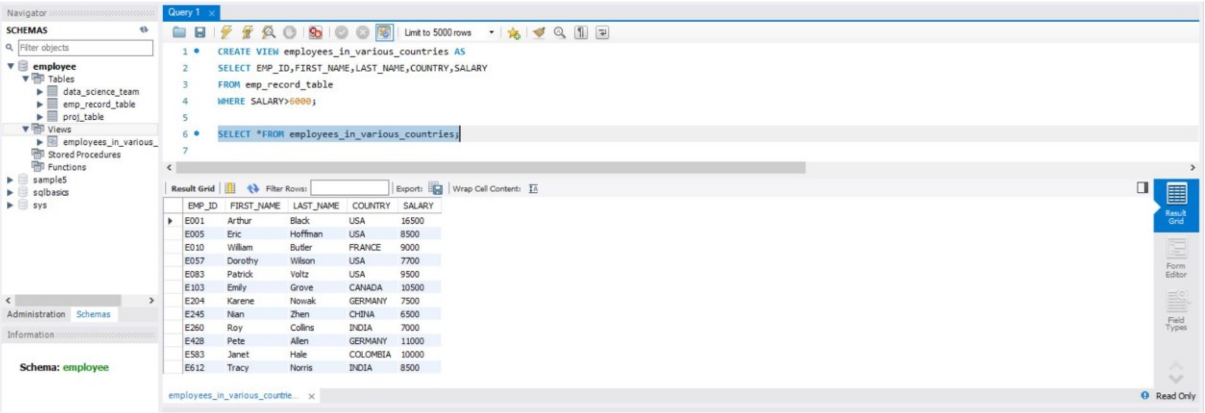
CREATE VIEW employees_in_various_countries AS

SELECT EMP_ID,FIRST_NAME,LAST_NAME,COUNTRY,SALARY

FROM emp_record_table

WHERE SALARY>6000;

SELECT *FROM employees_in_various_countries;



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

SELECT *FROM employees_in_various_countries;
```

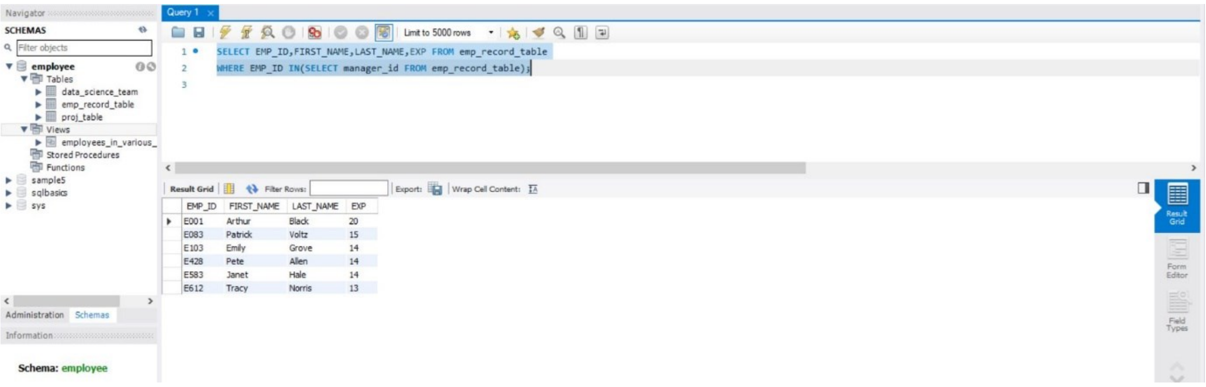
EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
E001	Arthur	Black	USA	16500
E005	Eric	Hoffman	USA	8500
E010	William	Butler	FRANCE	9000
E057	Dorothy	Wilson	USA	7700
E083	Patrick	Voltz	USA	9500
E103	Emily	Grove	CANADA	10500
E204	Karene	Nowak	GERMANY	7500
E245	Nian	Zhen	CHINA	6500
E260	Roy	Collins	INDIA	7000
E428	Pete	Allen	GERMANY	11000
E583	Janet	Hale	COLOMBIA	10000
E612	Tracy	Norris	INDIA	8500

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

12A.

SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP FROM emp_record_table

WHERE EMP_ID IN(SELECT manager_id FROM emp_record_table);



```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP FROM emp_record_table
WHERE EMP_ID IN(SELECT manager_id FROM emp_record_table);
```

EMP_ID	FIRST_NAME	LAST_NAME	EXP
E001	Arthur	Black	20
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.

13A.

DELIMITER &&

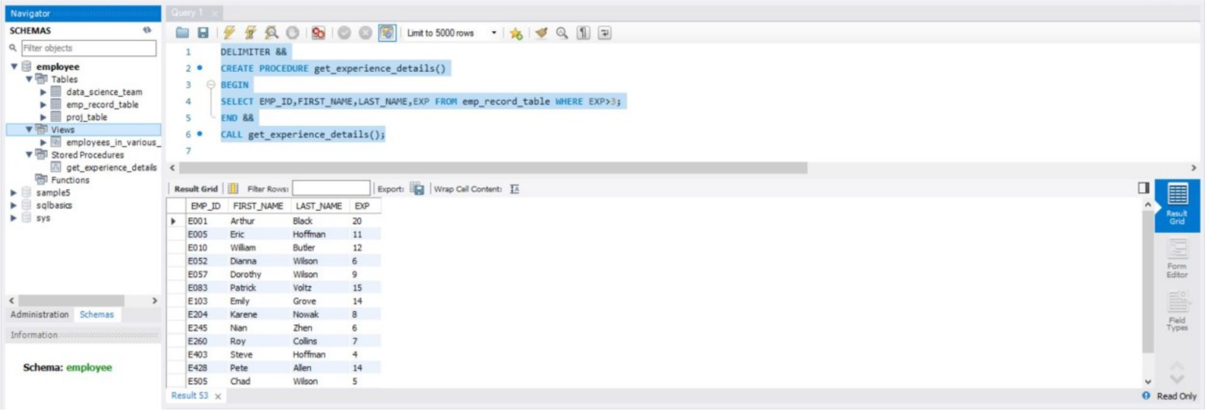
CREATE PROCEDURE get_experience_details()

BEGIN

SELECT EMP_ID,FIRST_NAME,LAST_NAME,EXP FROM emp_record_table WHERE EXP>3;

END &&

CALL get_experience_details();



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

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
E403	Steve	Hoffman	4
E428	Pete	Allen	14
E505	Chad	Wilson	5

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',

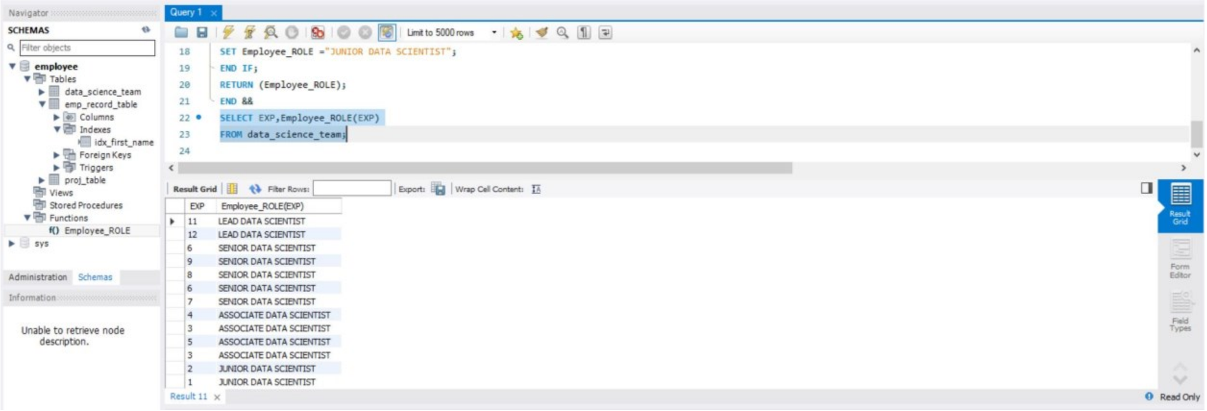
For an employee with the experience of 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST',

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'.

14A.

```
DELIMITER &&
CREATE FUNCTION Employee_ROLE(
EXP int
)
RETURNS VARCHAR(40)
DETERMINISTIC
BEGIN
DECLARE Employee_ROLE VARCHAR(40);
IF EXP>12 AND 16 THEN
SET Employee_ROLE="MANAGER";
ELSEIF EXP>10 AND 12 THEN
SET Employee_ROLE ="LEAD DATA SCIENTIST";
ELSEIF EXP>5 AND 10 THEN
SET Employee_ROLE ="SENIOR DATA SCIENTIST";
ELSEIF EXP>2 AND 5 THEN
SET Employee_ROLE ="ASSOCIATE DATA SCIENTIST";
ELSEIF EXP<=2 THEN
SET Employee_ROLE ="JUNIOR DATA SCIENTIST";
END IF;
RETURN (Employee_ROLE);
END &&

SELECT EXP,Employee_ROLE(EXP)
FROM data_science_team;
```



```
SET Employee_ROLE ="JUNIOR DATA SCIENTIST";
END IF;
RETURN (Employee_ROLE);
END &&
SELECT EXP,Employee_ROLE(EXP)
FROM data_science_team;
```

EXP	Employee_ROLE(EXP)
11	LEAD DATA SCIENTIST
12	LEAD DATA SCIENTIST
6	SENIOR DATA SCIENTIST
9	SENIOR DATA SCIENTIST
8	SENIOR DATA SCIENTIST
6	SENIOR DATA SCIENTIST
7	SENIOR DATA SCIENTIST
4	ASSOCIATE DATA SCIENTIST
3	ASSOCIATE DATA SCIENTIST
5	ASSOCIATE DATA SCIENTIST
3	ASSOCIATE DATA SCIENTIST
2	JUNIOR DATA SCIENTIST
1	JUNIOR DATA SCIENTIST

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.

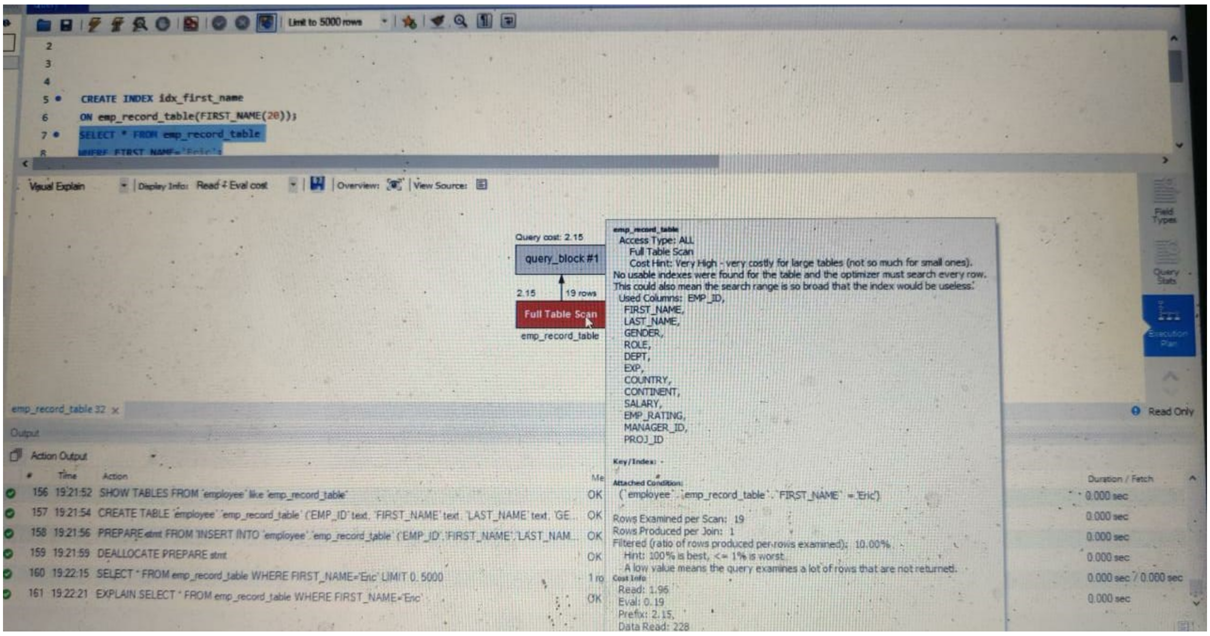
15A.

CREATE INDEX idx_first_name

ON emp_record_table(FIRST_NAME(20));

SELECT * FROM emp_record_table

WHERE FIRST_NAME='Eric';



```
CREATE INDEX idx_first_name
ON emp_record_table(FIRST_NAME(20));
SELECT * FROM emp_record_table
WHERE FIRST_NAME='Eric';
```

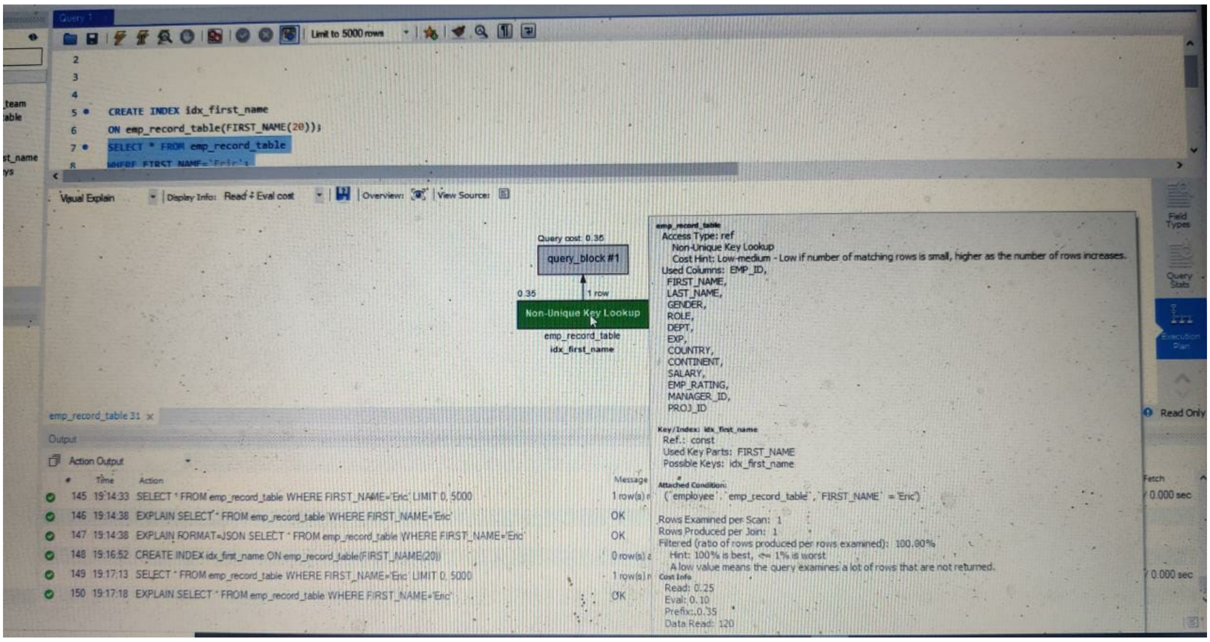
```
emp_record_table
 Access Type: ALL
   Full Table Scan
   Cost Hint: Very High - very costly for large tables (not so much for small ones).
No usable indexes were found for the table and the optimizer must search every row.
This could also mean the search range is so broad that the index would be useless.
 Used Columns: EMP_ID,
  FIRST_NAME,
  LAST_NAME,
  GENDER,
  ROLE,
  DEPT,
  EXP,
  COUNTRY,
  CONTINENT,
  SALARY,
  EMP_RATING,
  MANAGER_ID,
  PROJ_ID

Key/Index: -

Attached Condition:
 (`employee`.`emp_record_table`.`FIRST_NAME` = 'Eric')

Rows Examined per Scan: 19
Rows Produced per Join: 1
Filtered (ratio of rows produced per rows examined): 10.00%
  Hint: 100% is best, <= 1% is worst
  A low value means the query examines a lot of rows that are not returned.
Cost Info
 Read: 1.96
 Eval: 0.19
 Prefix: 2.15,
 Data Read: 228
```

```
#    Time      Action                                                                                          Me    Duration / Fetch
156  19:21:52  SHOW TABLES FROM `employee` like 'emp_record_table'                                             OK    0.000 sec
157  19:21:54  CREATE TABLE `employee`.`emp_record_table` (`EMP_ID` text, `FIRST_NAME` text, `LAST_NAME` text, `GE...  OK    0.000 sec
158  19:21:56  PREPARE stmt FROM 'INSERT INTO `employee`.`emp_record_table` (`EMP_ID`,`FIRST_NAME`,`LAST_NAM...  OK    0.000 sec
159  19:21:59  DEALLOCATE PREPARE stmt                                                                         OK    0.000 sec
160  19:22:15  SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric' LIMIT 0, 5000                            1 ro  0.000 sec / 0.000 sec
161  19:22:21  EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'                                  OK    0.000 sec
```



```
CREATE INDEX idx_first_name
ON emp_record_table(FIRST_NAME(20));
SELECT * FROM emp_record_table
WHERE FIRST_NAME='Eric';
```

```
emp_record_table
  Access Type: ref
    Non-Unique Key Lookup
    Cost Hint: Low-medium - Low if number of matching rows is small, higher as the number of rows increases.
  Used Columns: EMP_ID,
   FIRST_NAME,
   LAST_NAME,
   GENDER,
   ROLE,
   DEPT,
   EXP,
   COUNTRY,
   CONTINENT,
   SALARY,
   EMP_RATING,
   MANAGER_ID,
   PROJ_ID
Key/Index: idx_first_name
  Ref.: const
  Used Key Parts: FIRST_NAME
  Possible Keys: idx_first_name
Attached Condition:
  (`employee`.`emp_record_table`.`FIRST_NAME` = 'Eric')

Rows Examined per Scan: 1
Rows Produced per Join: 1
Filtered (ratio of rows produced per rows examined): 100.00%
   Hint: 100% is best, <= 1% is worst
   A low value means the query examines a lot of rows that are not returned.
Cost Info
  Read: 0.25
  Eval: 0.10
  Prefix: 0.35
  Data Read: 120
```

```
Action Output
    #   Time      Action                                                                          Message
✓  145  19:14:33  SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric' LIMIT 0, 5000              1 row(s) r
✓  146  19:14:38  EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'                    OK
✓  147  19:14:38  EXPLAIN FORMAT=JSON SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'        OK
✓  148  19:16:52  CREATE INDEX idx_first_name ON emp_record_table(FIRST_NAME(20))                   0 row(s) a
✓  149  19:17:13  SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric' LIMIT 0, 5000              1 row(s) r
✓  150  19:17:18  EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME='Eric'                    OK
```

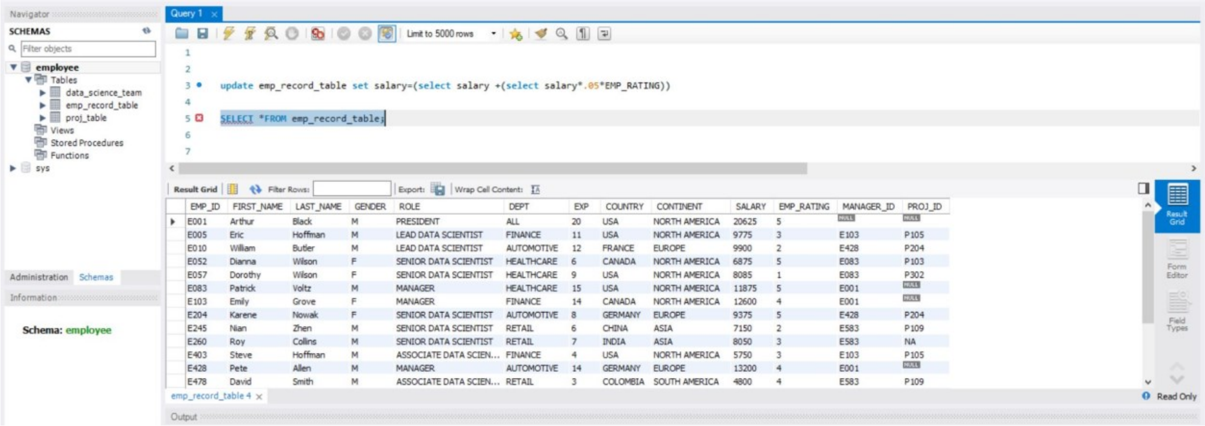
Fetch
0.000 sec
0.000 sec

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).

16A.

update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING))

SELECT *FROM emp_record_table;



```
update emp_record_table set salary=(select salary +(select salary*.05*EMP_RATING))

SELECT *FROM emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	20625	5	NULL	NULL
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	9775	3	E103	P105
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9900	2	E428	P204
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	6875	5	E083	P103
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	8085	1	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	11875	5	E001	NULL
E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	12600	4	E001	NULL
E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	9375	5	E428	P204
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	7150	2	E583	P109
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	8050	3	E583	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIEN...	FINANCE	4	USA	NORTH AMERICA	5750	3	E103	P105
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	13200	4	E001	NULL
E478	David	Smith	M	ASSOCIATE DATA SCIEN...	RETAIL	3	COLOMBIA	SOUTH AMERICA	4800	4	E583	P109

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

17A.

SELECT EMP_ID,FIRST_NAME,LAST_NAME,SALARY,COUNTRY,CONTINENT,

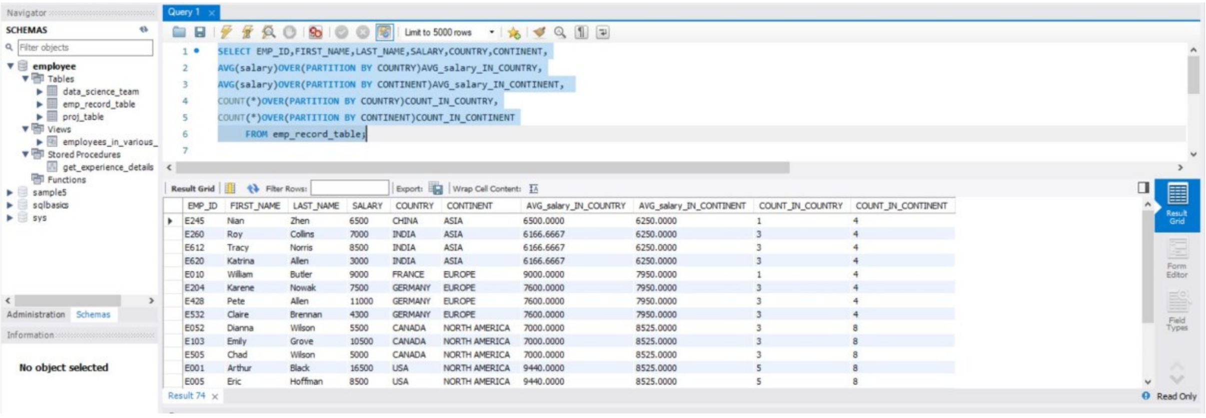
AVG(salary)OVER(PARTITION BY COUNTRY)AVG_salary_IN_COUNTRY,

AVG(salary)OVER(PARTITION BY CONTINENT)AVG_salary_IN_CONTINENT,

COUNT(*)OVER(PARTITION BY COUNTRY)COUNT_IN_COUNTRY,

COUNT(*)OVER(PARTITION BY CONTINENT)COUNT_IN_CONTINENT

FROM emp_record_table;



```
SELECT EMP_ID,FIRST_NAME,LAST_NAME,SALARY,COUNTRY,CONTINENT,
AVG(salary)OVER(PARTITION BY COUNTRY)AVG_salary_IN_COUNTRY,
AVG(salary)OVER(PARTITION BY CONTINENT)AVG_salary_IN_CONTINENT,
COUNT(*)OVER(PARTITION BY COUNTRY)COUNT_IN_COUNTRY,
COUNT(*)OVER(PARTITION BY CONTINENT)COUNT_IN_CONTINENT
    FROM emp_record_table;
```

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY	CONTINENT	AVG_salary_IN_COUNTRY	AVG_salary_IN_CONTINENT	COUNT_IN_COUNTRY	COUNT_IN_CONTINENT
E245	Nian	Zhen	6500	CHINA	ASIA	6500.0000	6250.0000	1	4
E260	Roy	Collins	7000	INDIA	ASIA	6166.6667	6250.0000	3	4
E612	Tracy	Norris	8500	INDIA	ASIA	6166.6667	6250.0000	3	4
E620	Katrina	Allen	3000	INDIA	ASIA	6166.6667	6250.0000	3	4
E010	William	Butler	9000	FRANCE	EUROPE	9000.0000	7950.0000	1	4
E204	Karene	Nowak	7500	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E428	Pete	Allen	11000	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E532	Claire	Brennan	4300	GERMANY	EUROPE	7600.0000	7950.0000	3	4
E052	Dianna	Wilson	5500	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E103	Emily	Grove	10500	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E505	Chad	Wilson	5000	CANADA	NORTH AMERICA	7000.0000	8525.0000	3	8
E001	Arthur	Black	16500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8
E005	Eric	Hoffman	8500	USA	NORTH AMERICA	9440.0000	8525.0000	5	8