Scienceqtech Employee Performance Mapping

SQL PROJECT

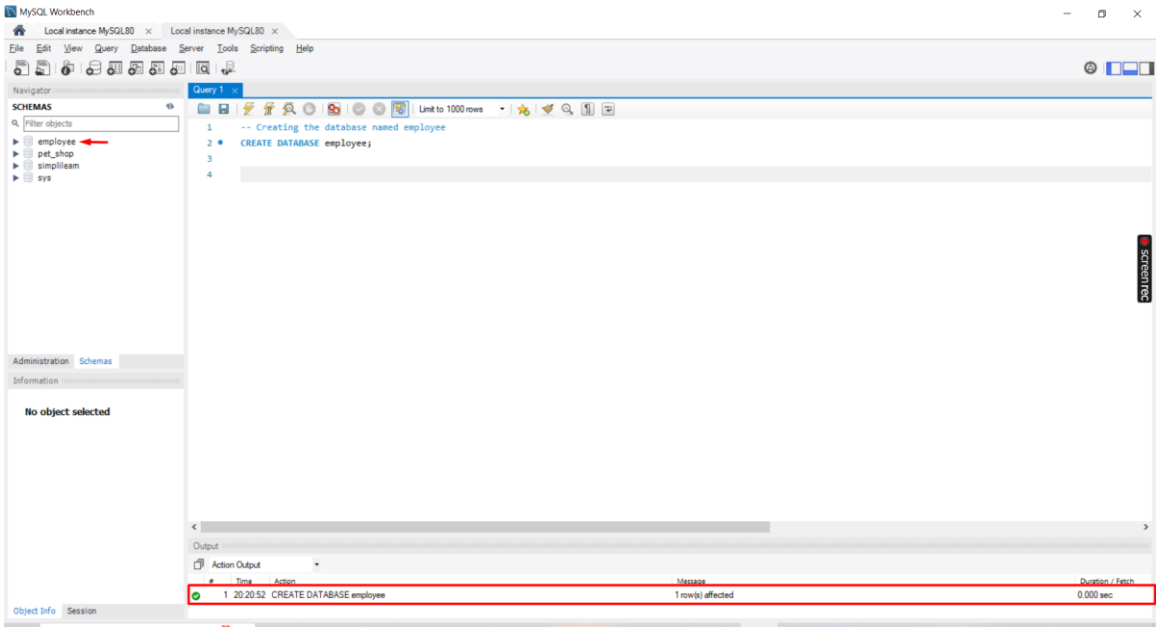
VANDANA T V | IITK DATASCIENCE | 30/07/2024

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 a database ‘employee’ using the query:

```
CREATE DATABASE employee;
```

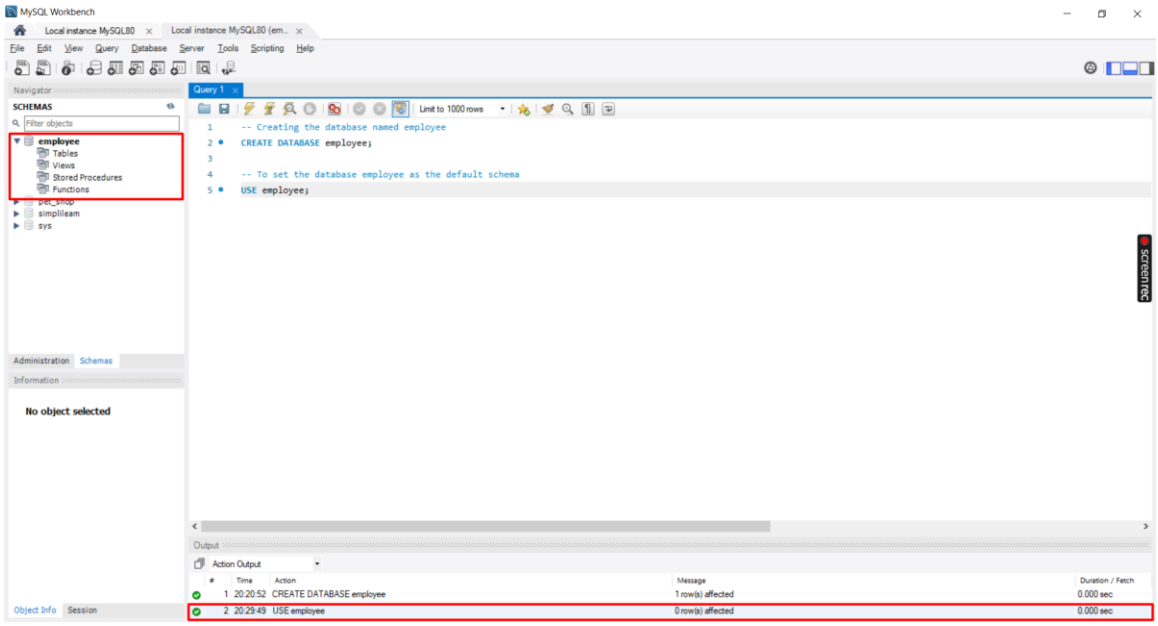
Output:



To select the database ‘employee’ as a default database schema and perform the operations in it we have the query:

```
USE employee;
```

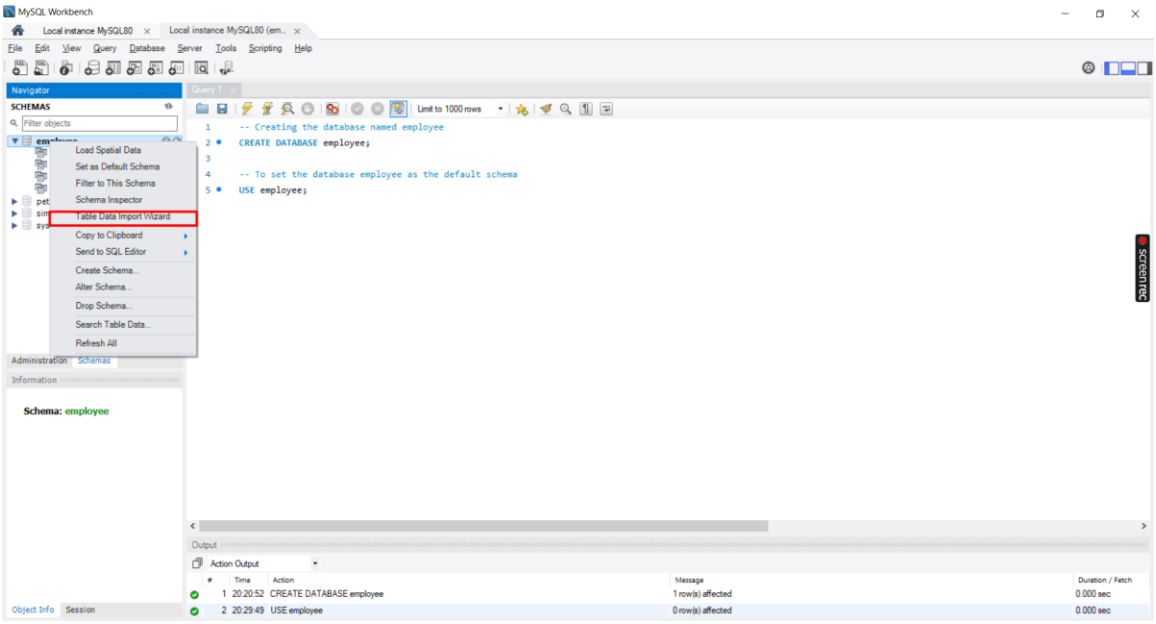
Output:


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
employee
Tables
Views
Stored Procedures
Functions
simplilearn
sys
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
-- Creating the database named employee
CREATE DATABASE employee;
-- To set the database employee as the default schema
USE employee;
Output
Action Output
Time Action Message Duration / Fetch
1 20:20:52 CREATE DATABASE employee 1 row(s) affected 0.000 sec
2 20:29:49 USE employee 0 row(s) affected 0.000 sec

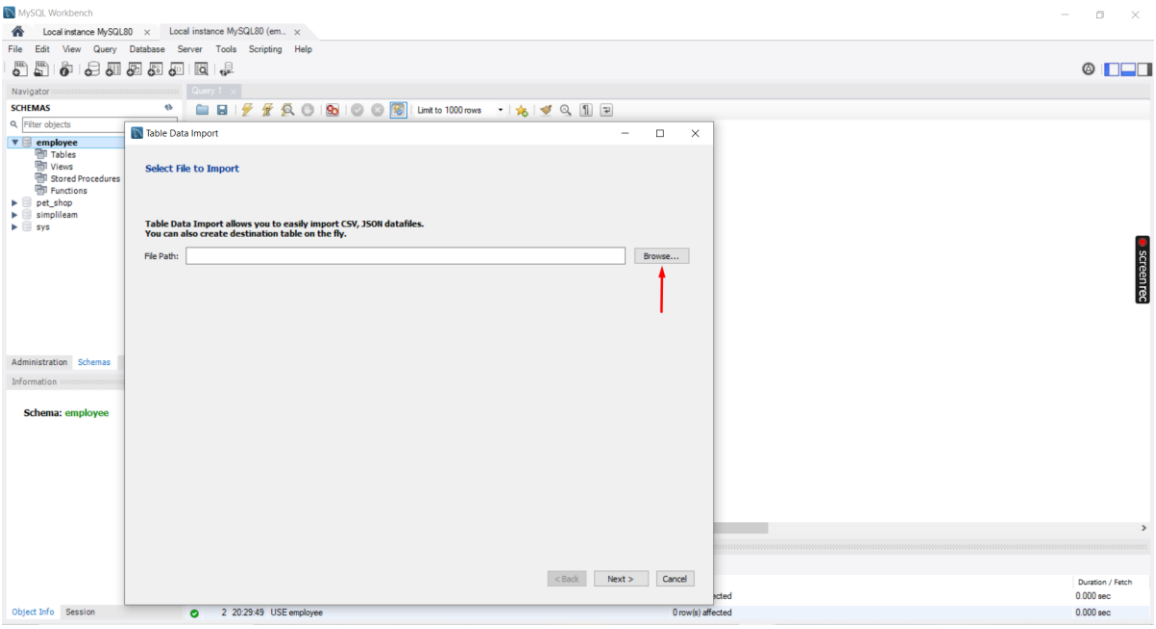
Importing Tables to the database:

Following are the steps to import the table data_science_team to the database employee

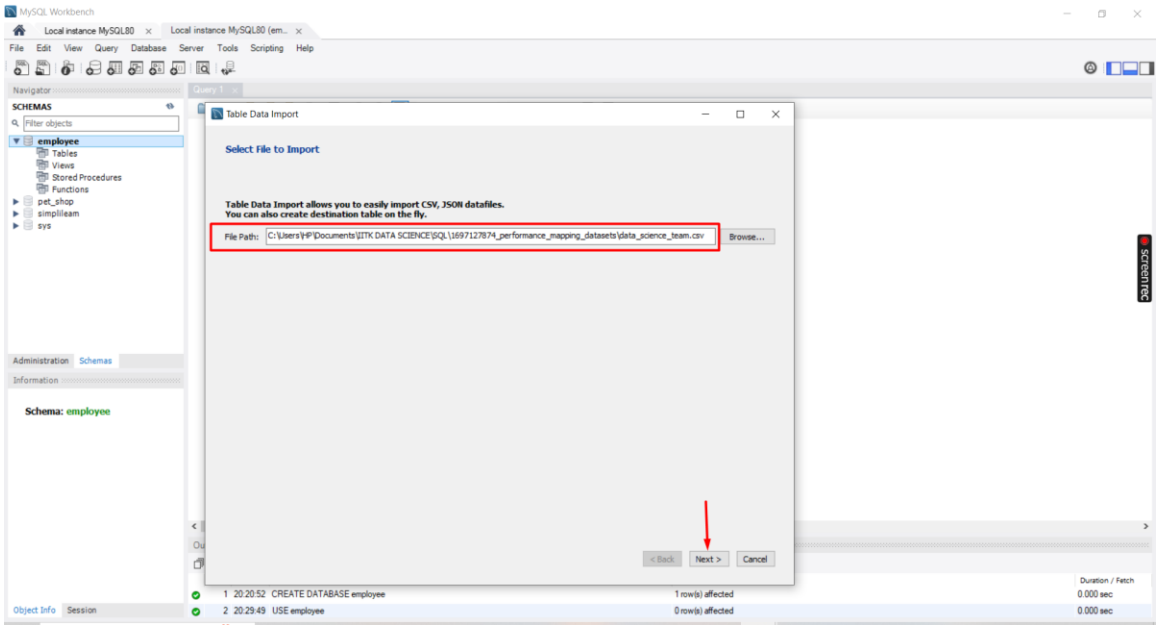
Step 1 : Right click on the employee database and choose ‘Table Data Import Wizard’ as shown below:



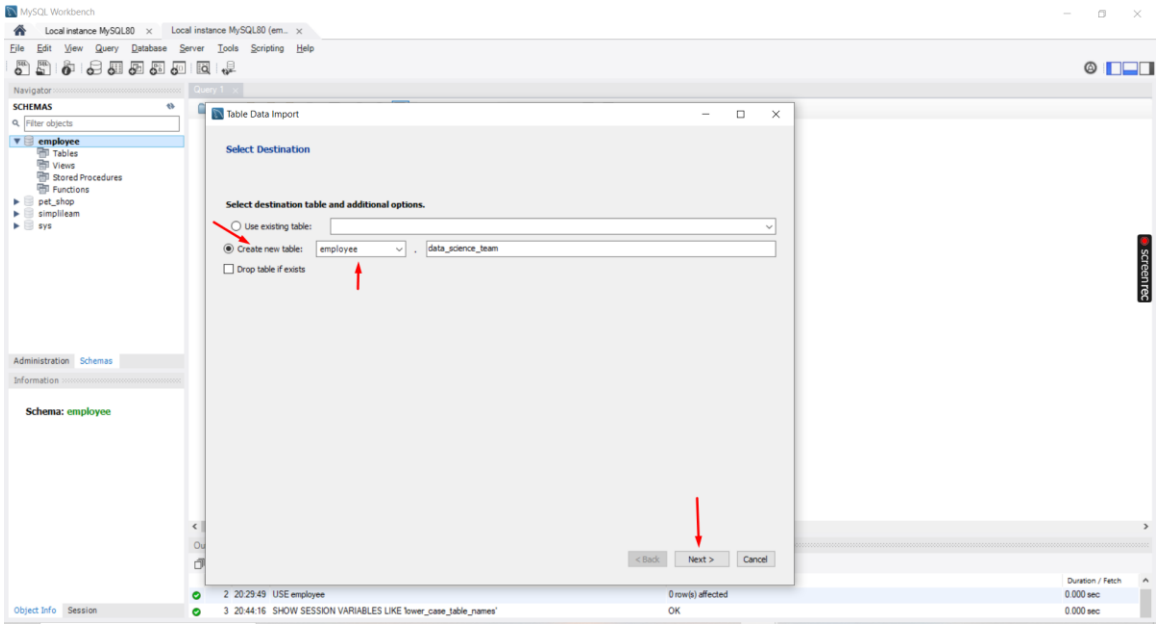
Step 2: Click on ‘Browse’ to select the file data_science_team.csv from the right location where the .csv file is saved.



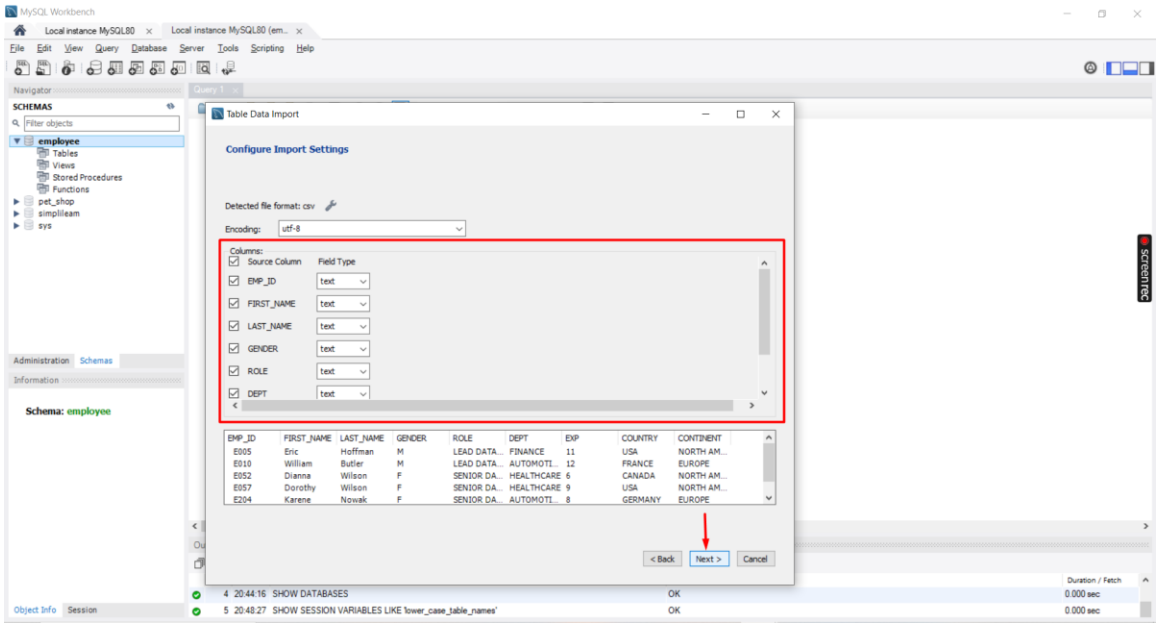
Step 3: Select the file data_science_team.csv from the saved location and click Next.



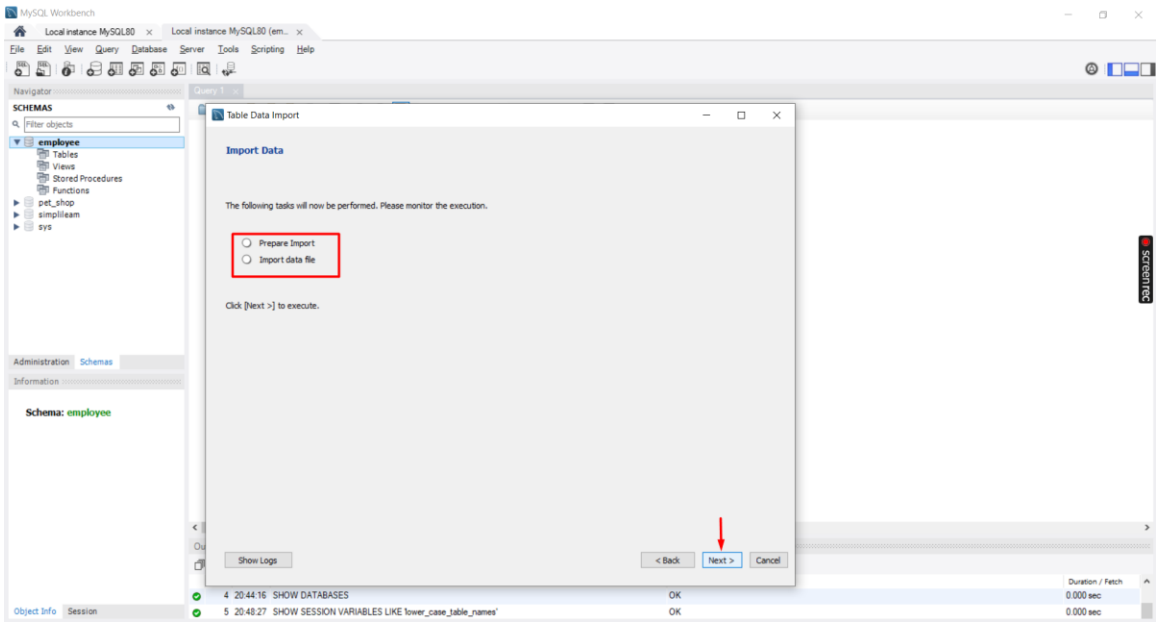
Step 4: Select ‘Create new table’ and make sure that the database is ‘employee’ then click Next.



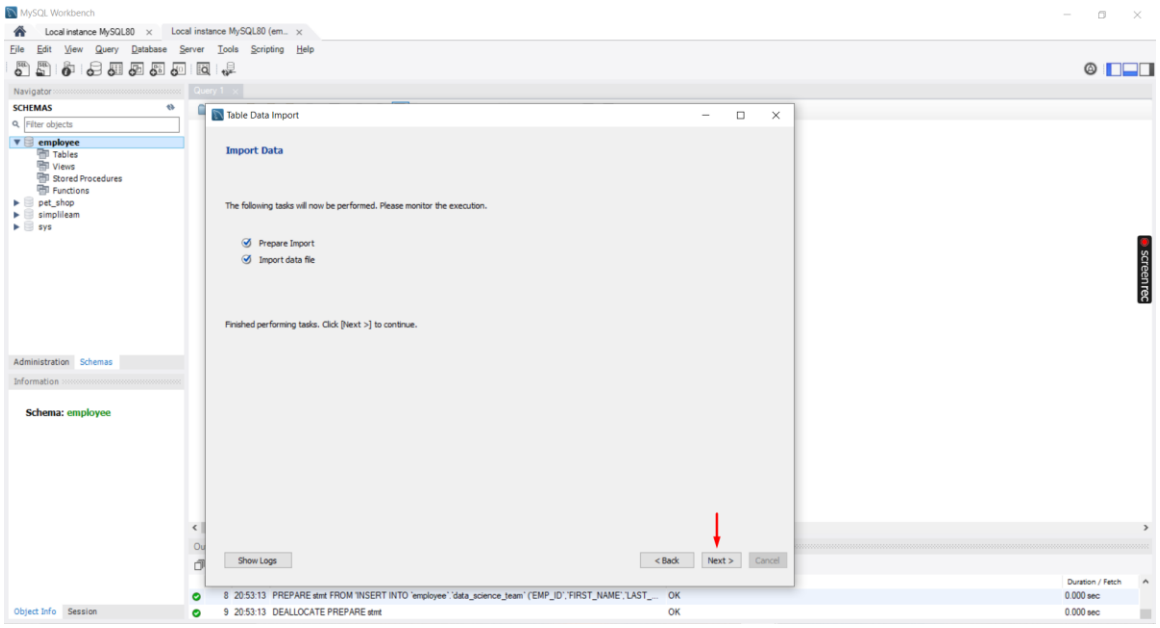
Step 5: Verify all the datatypes from the table which will be imported and click on Next.



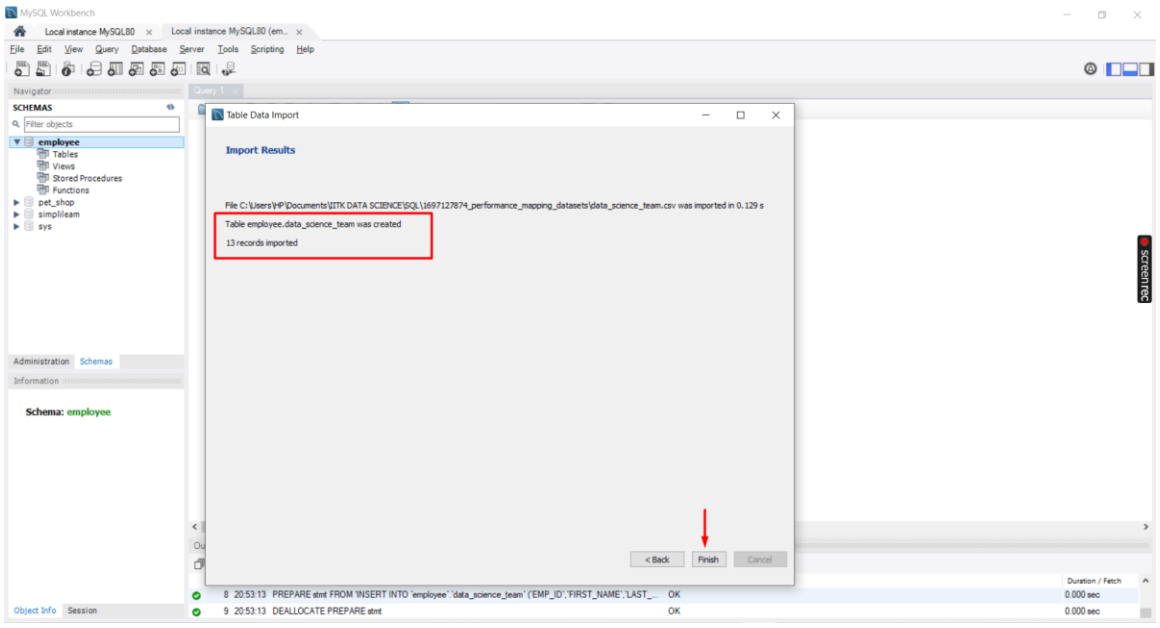
Step 6: In this window, we can see the steps which will be performed while importing the table and click Next which will start importing the table.



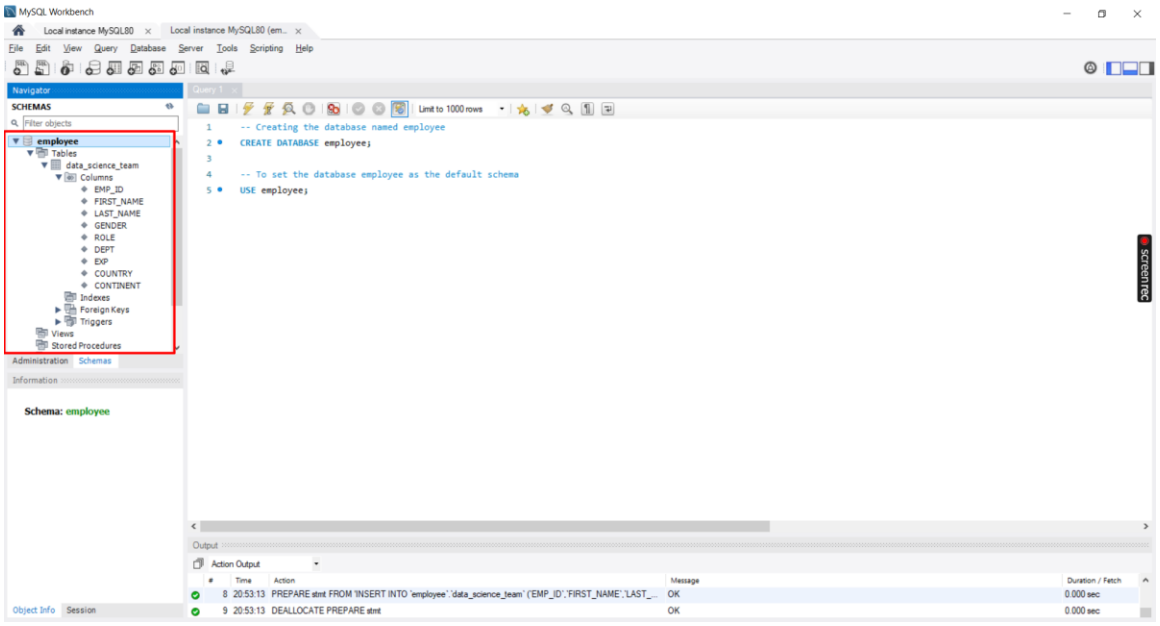
Step 7: Click Next in this window.



Step 8: Here in the below pop up window we can see the table has been created with 13 records. Click Finish to complete the creation of table named data_science_team.

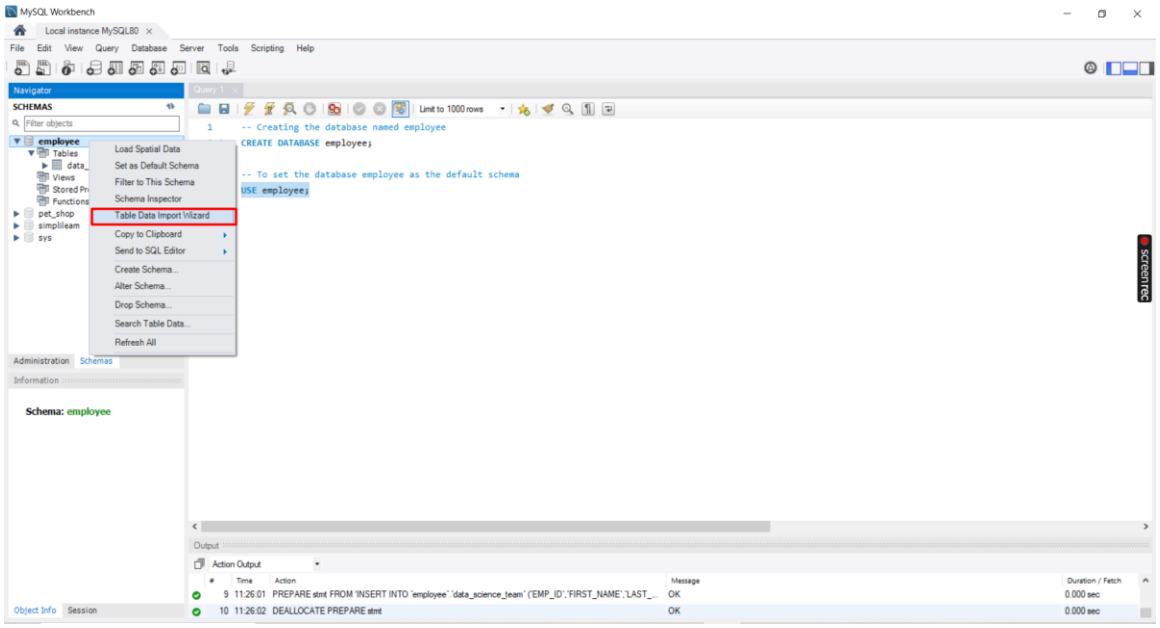


After finishing all the steps we can see the table ‘data_science_team’ in the database schema ‘employee’.

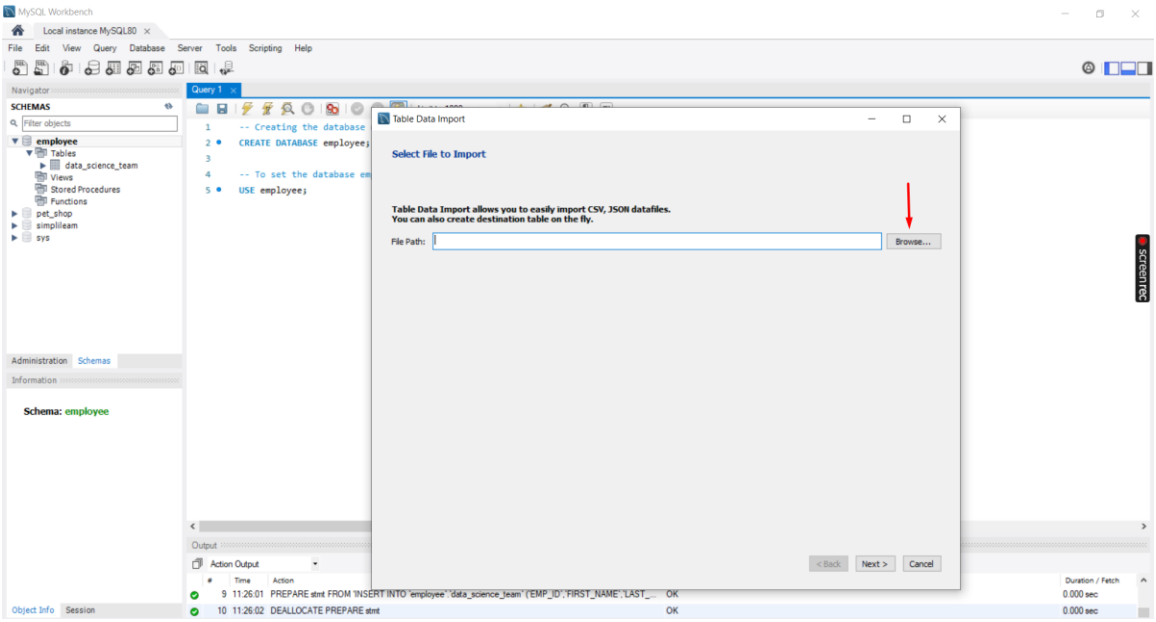


Following are the steps to import the table ‘emp_record_table’ to the database ‘employee’.

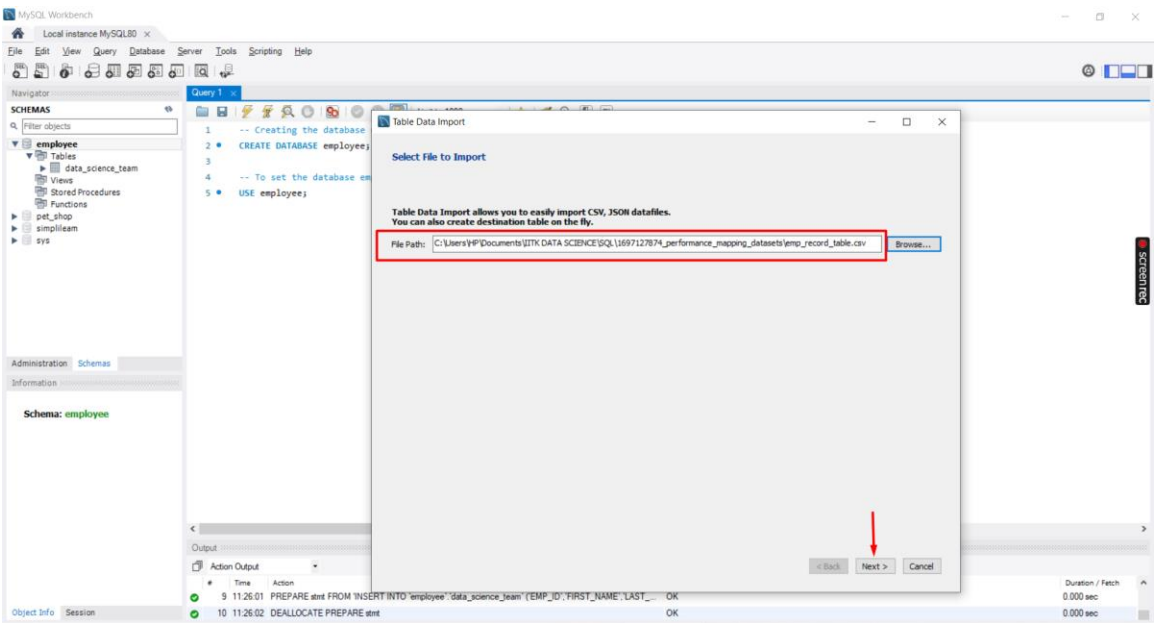
Step 1 : Right click on the employee database and choose ‘Table Data Import Wizard’ as shown below



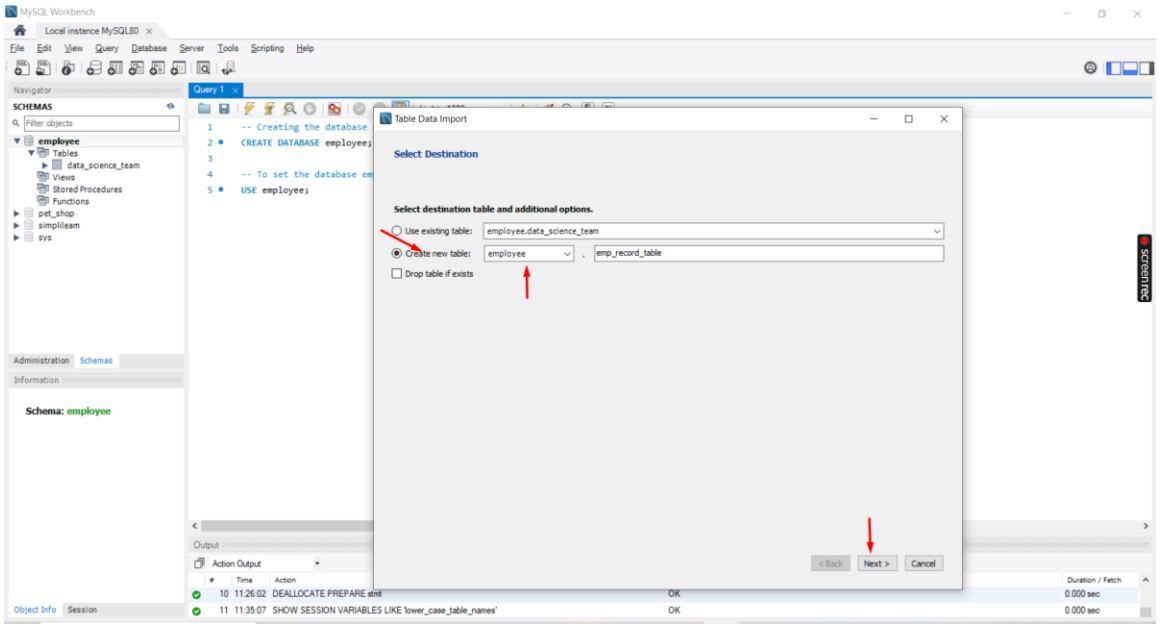
Step 2: Click on ‘Browse’ to select the file emp_record_table.csv from the right location where the .csv file is saved.



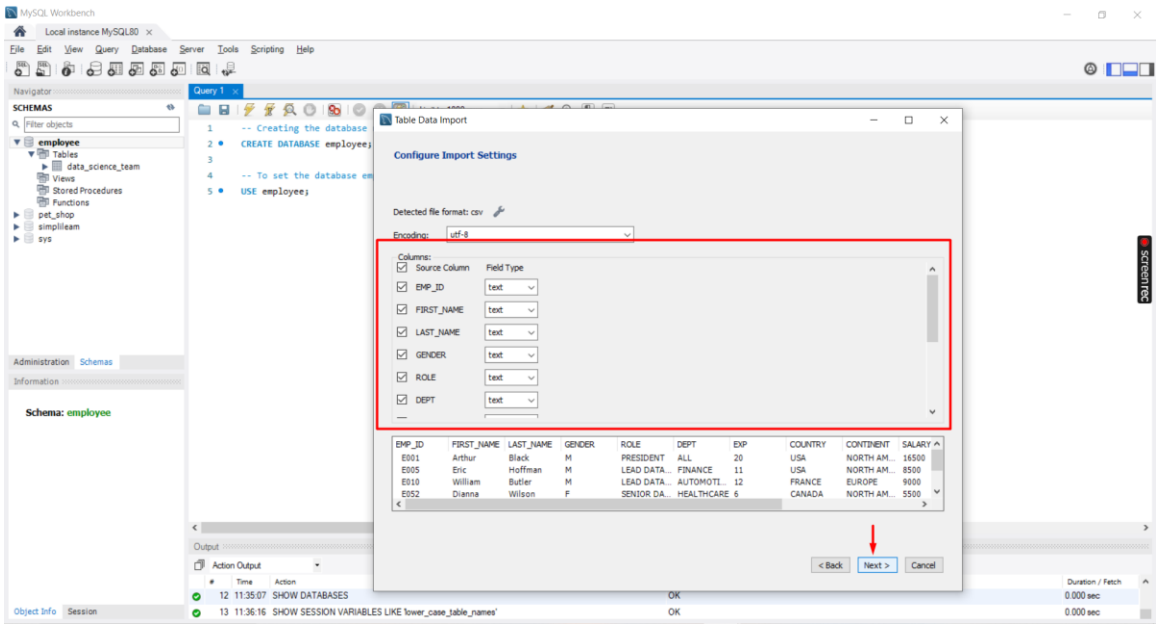
Step 3: Select the file emp_record_table.csv from the saved location and click Next.



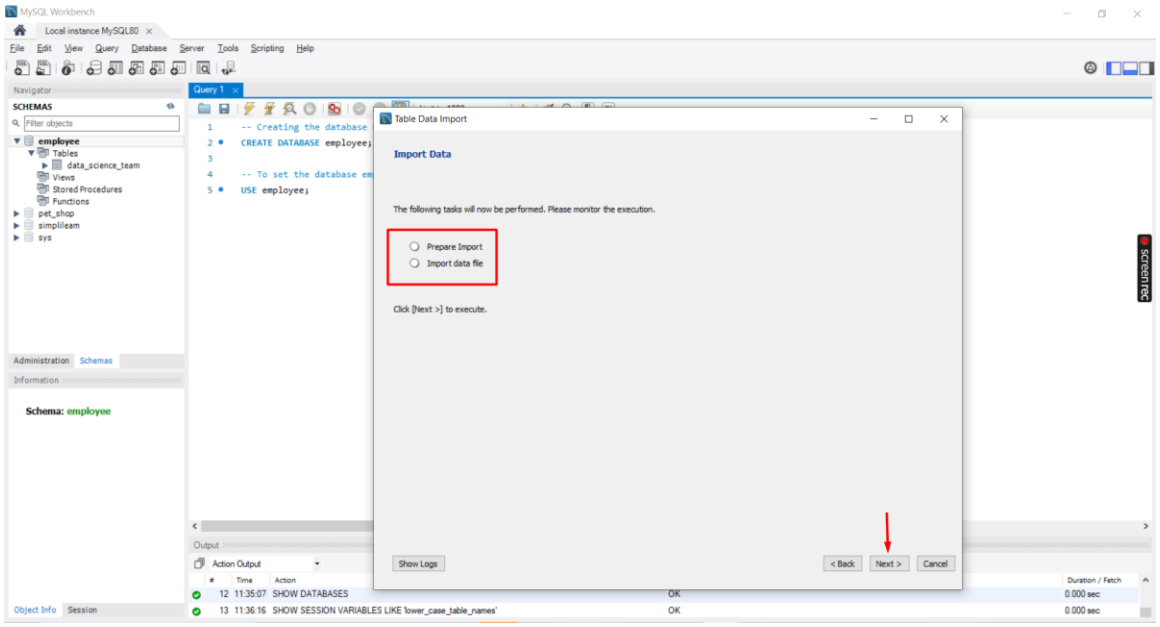
Step 4: Select ‘Create new table’ and make sure that the database is ‘employee’ then click Next.



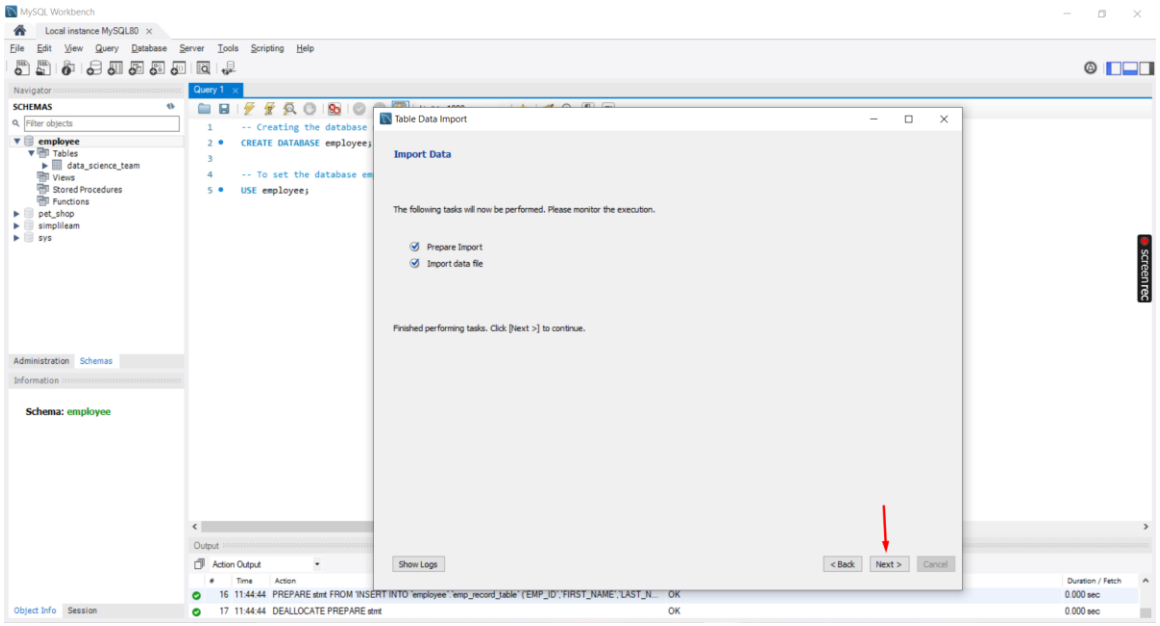
Step 5: Verify all the datatypes from the table which will be imported and click on Next.



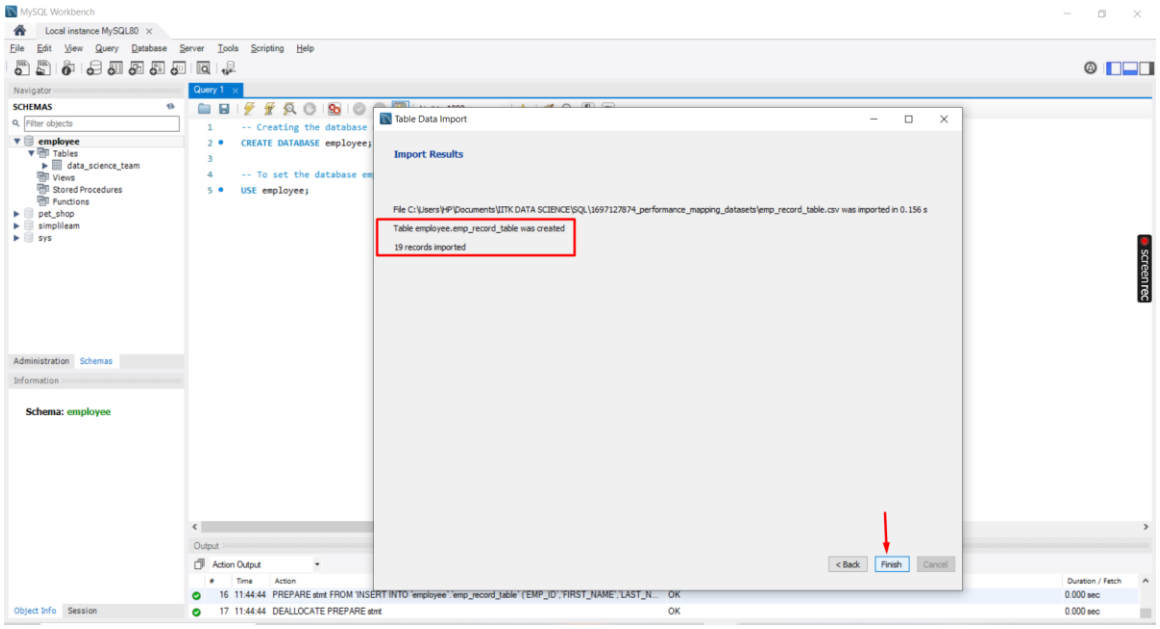
Step 6: In this window, we can see the steps which will be performed while importing the table and click Next which will start importing the table.



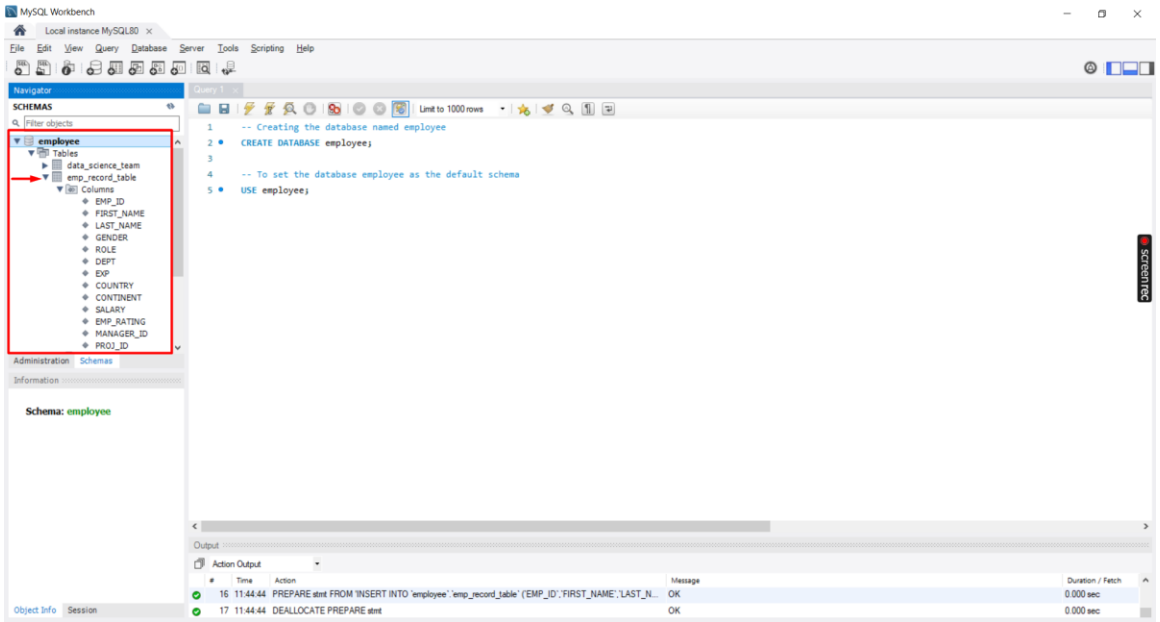
Step 7: Click Next in this window.



Step 8: Here in the below pop up window we can see the table has been created with 19 records. Click Finish to complete the creation of table named emp_record_table.

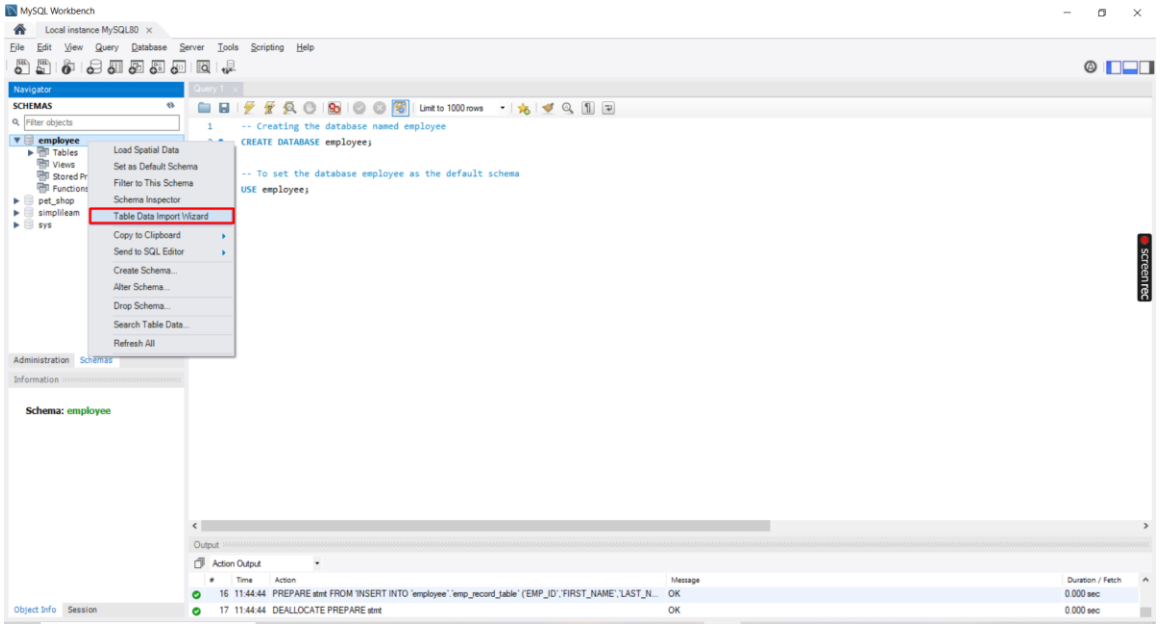


After finishing all the steps we can see the table ‘emp_record_table’ in the database schema ‘employee’.

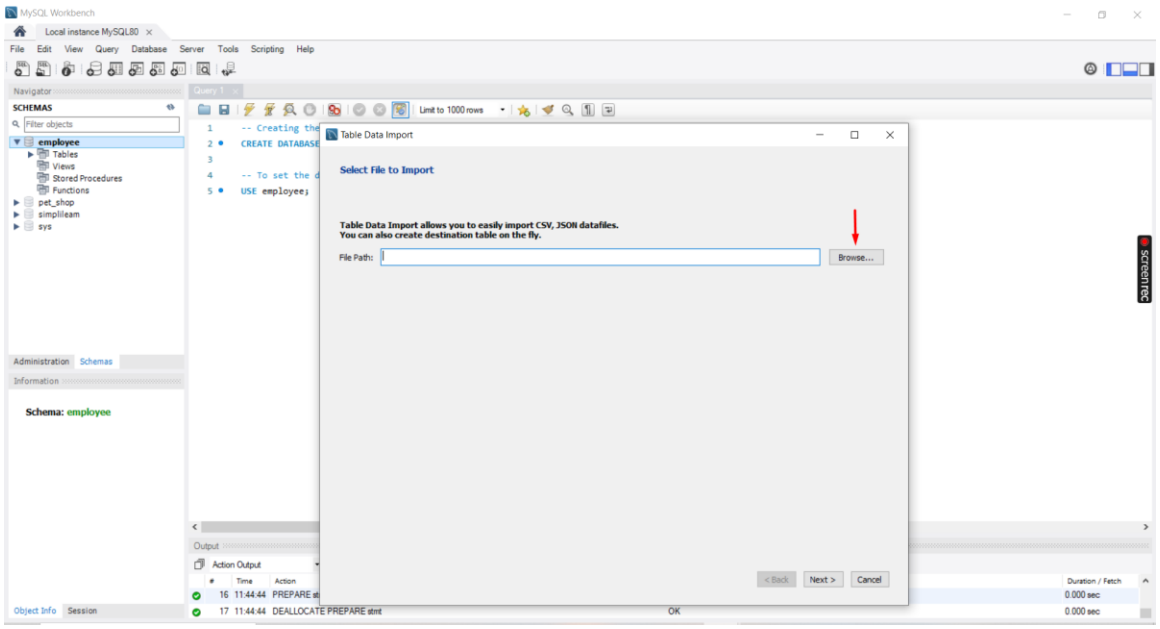


Following are the steps to import the table ‘proj_table’ to the database ‘employee’.

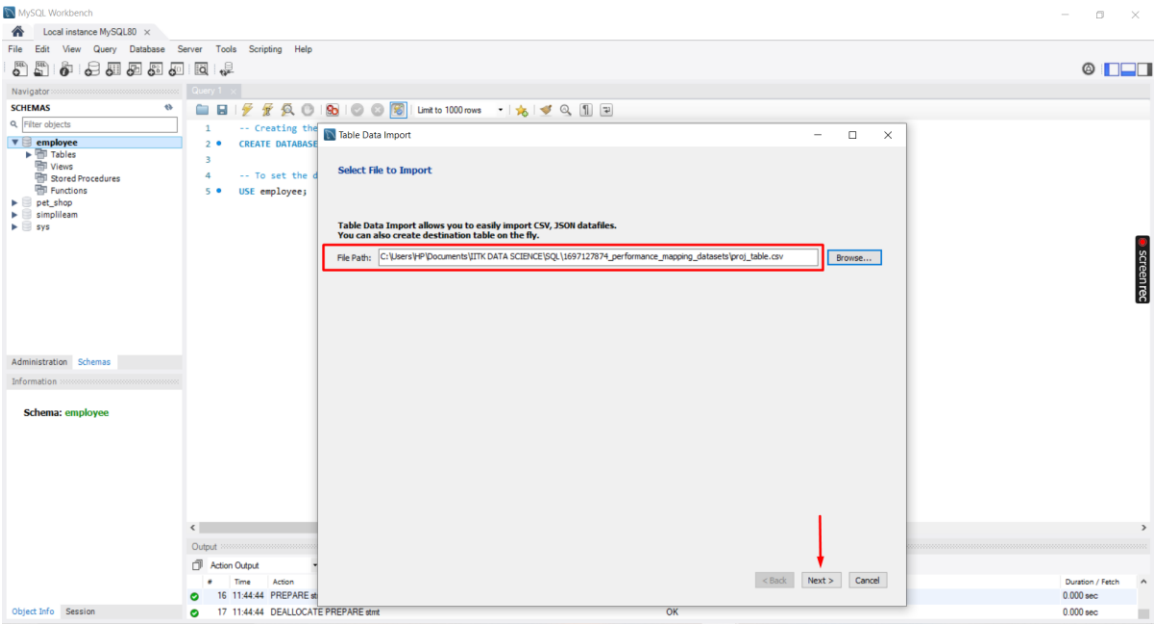
Step 1 : Right click on the employee database and choose ‘Table Data Import Wizard’ as shown below



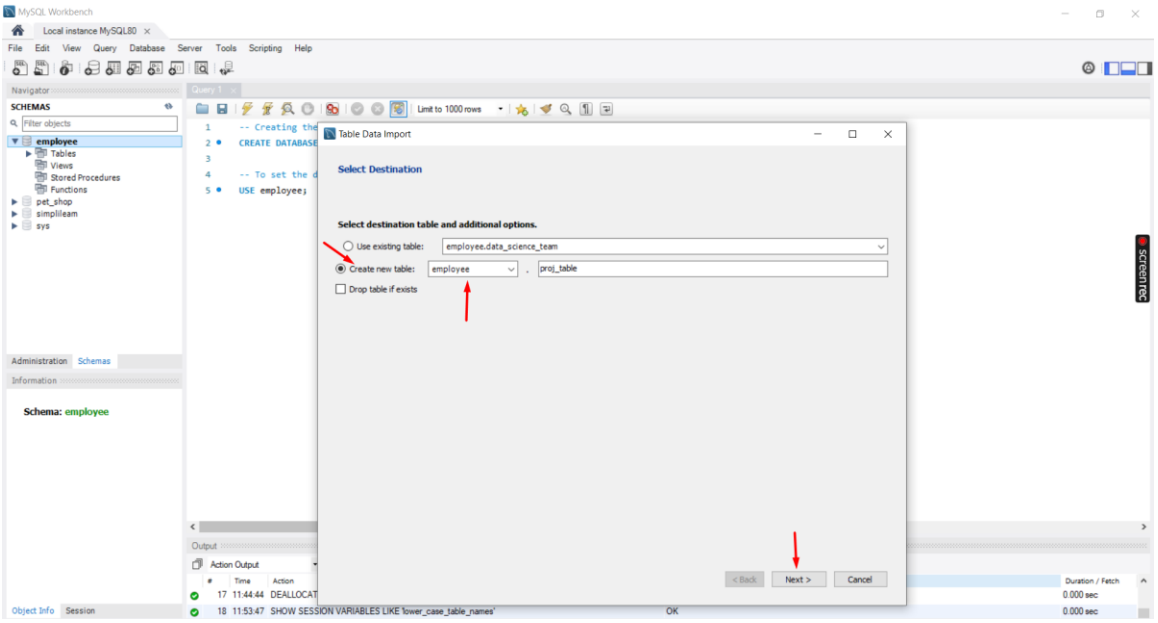
Step 2: Click on ‘Browse’ to select the file proj_table.csv from the right location where the .csv file is saved.



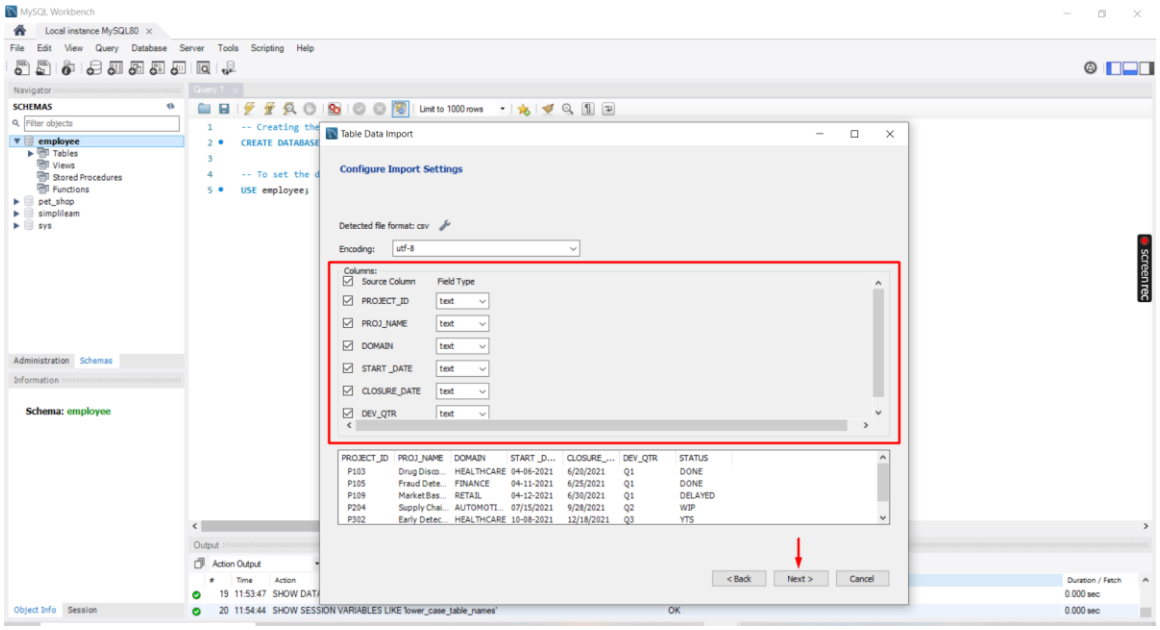
Step 3: Select the file proj_table.csv from the saved location and click Next.



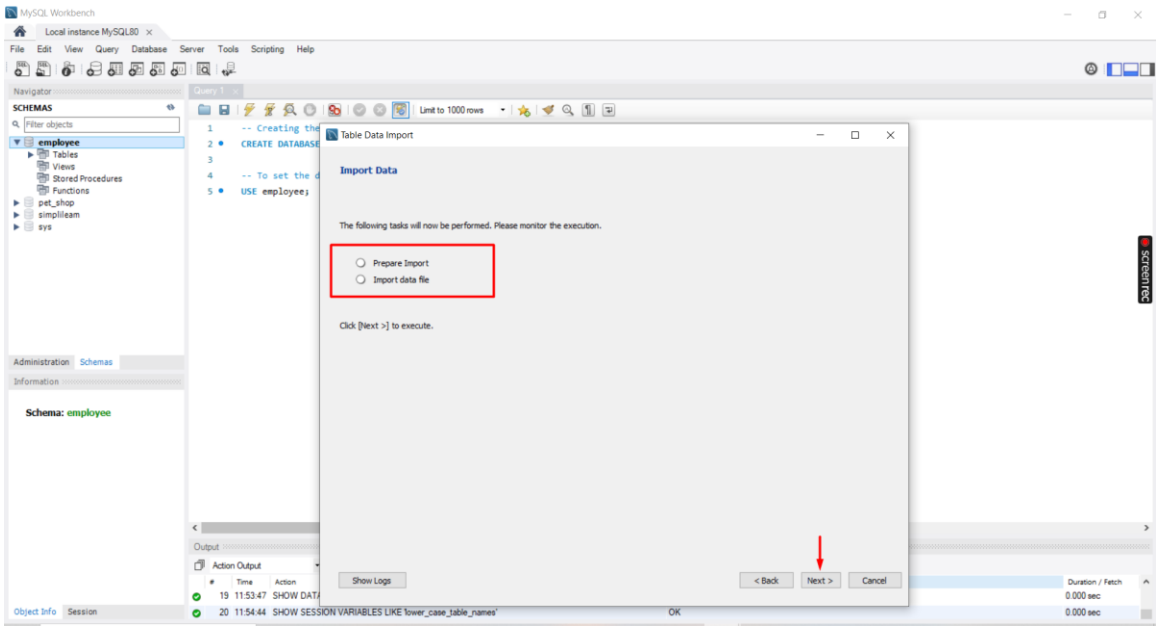
Step 4: Select ‘Create new table’ and make sure that the database is ‘employee’ then click Next.



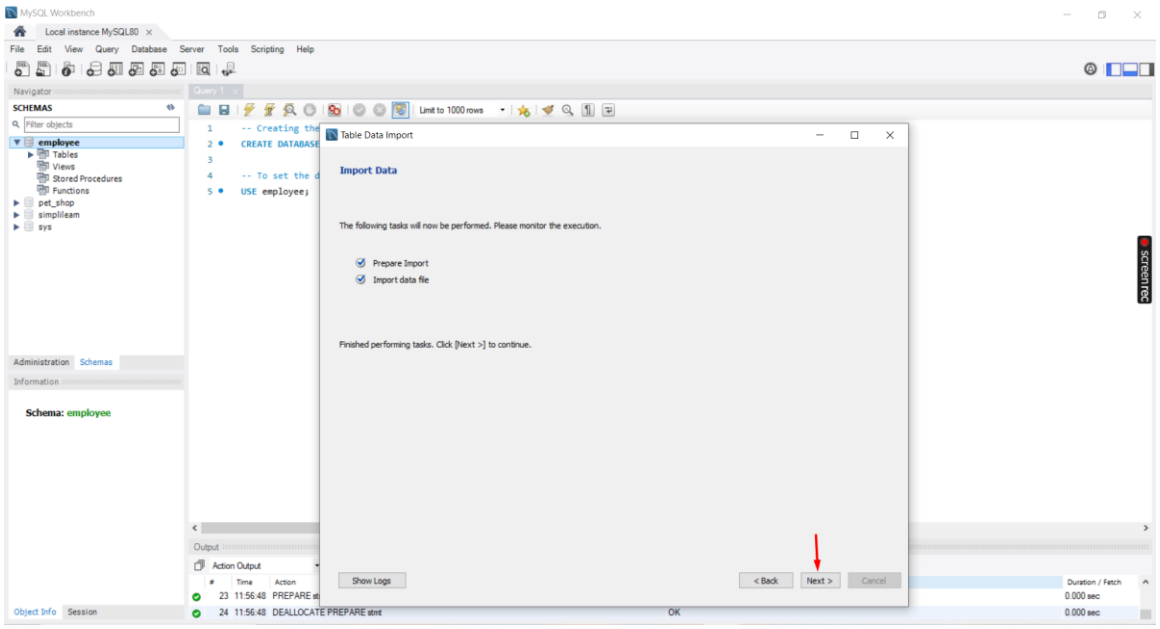
Step 5: Verify all the datatypes from the table which will be imported and click on Next.



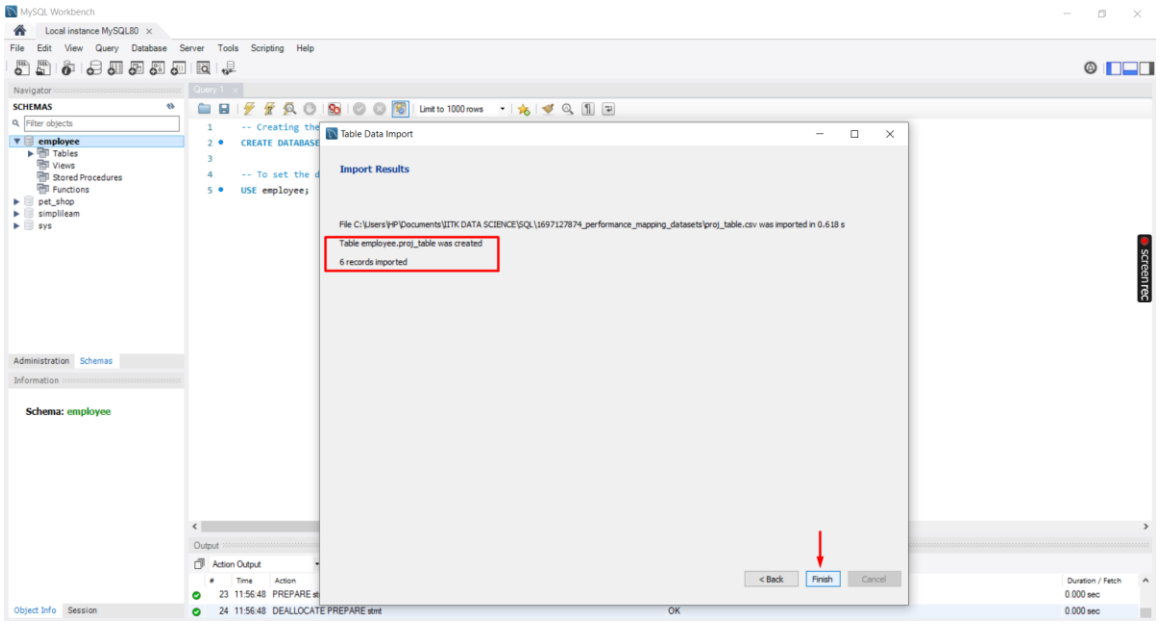
Step 6: In this window, we can see the steps which will be performed while importing the table and click Next which will start importing the table.



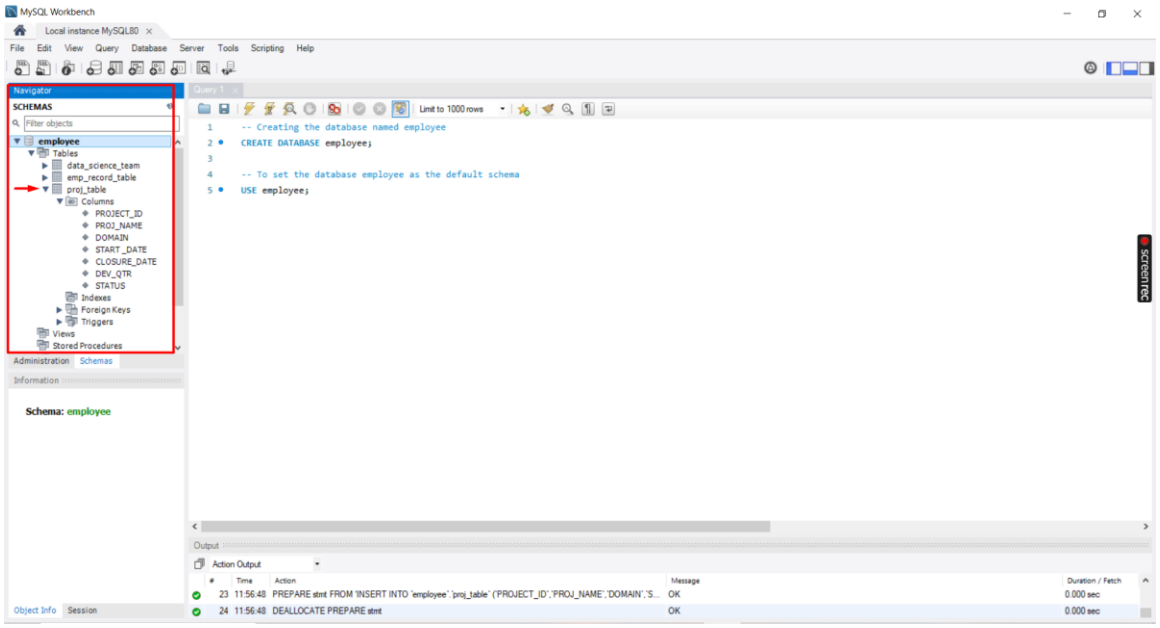
Step 7: Click Next in this window.



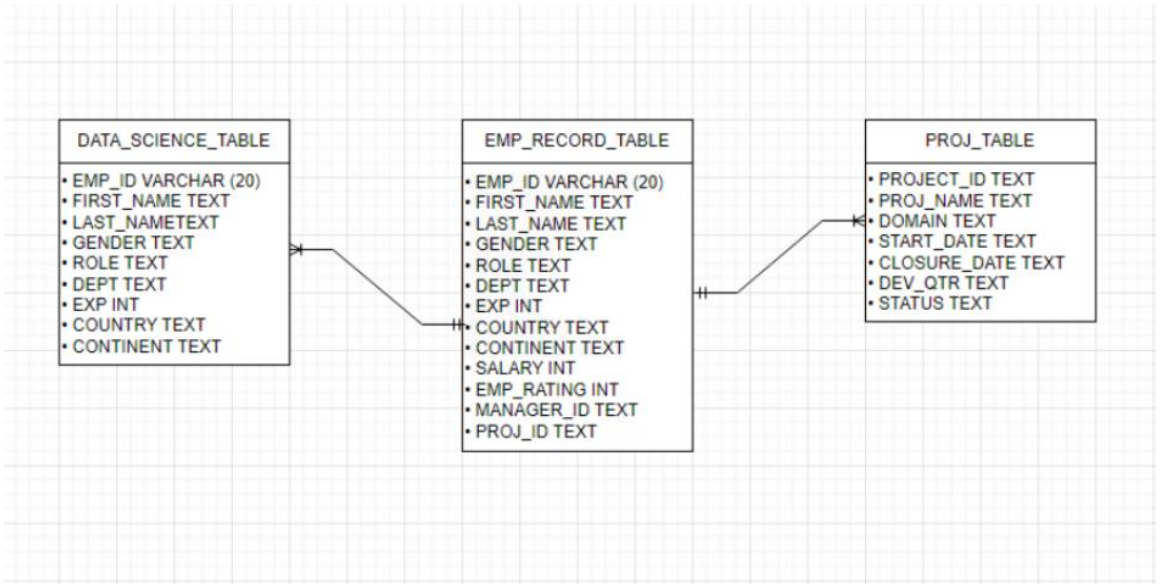
Step 8: Here in the below pop up window we can see the table has been created with 6 records. Click Finish to complete the creation of table named proj_table.



After finishing all the steps we can see the table ‘proj_table’ in the database schema ‘employee’.



2. Create an ER diagram for the given employee database

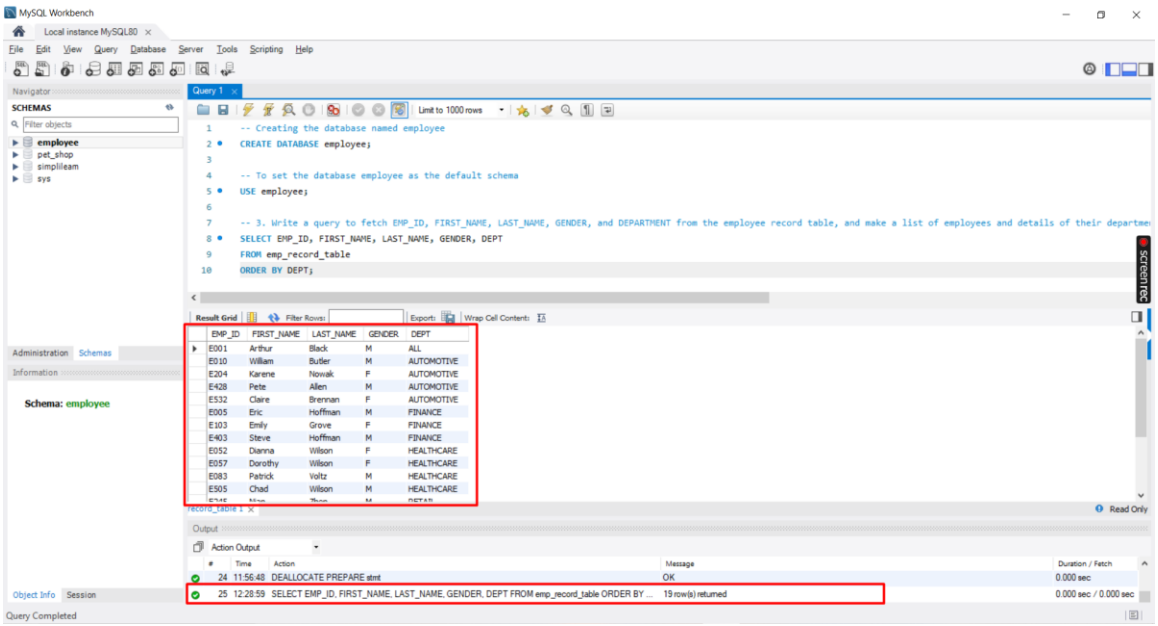


3. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

Here we have to retrieve the EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPT from the emp_record_table and group by the DEPT. For that execute the below query:

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

FROM emp_record_table

ORDER BY DEPT;
```

Result:



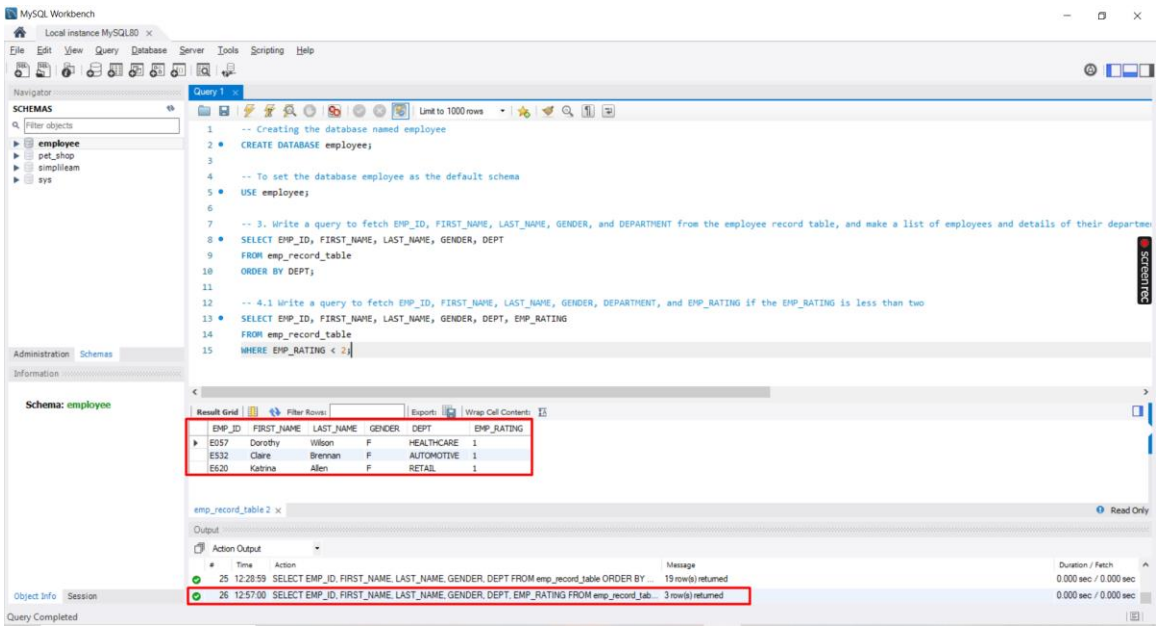
4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:
   - less than two
   - greater than four
   - between two and four

4.1: If the EMP_RATING is less than two

We can retrieve EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT and EMP_RATING whose EMP_RATING is less than 2 from the emp_record_table using the below query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING < 2;
```

Result:

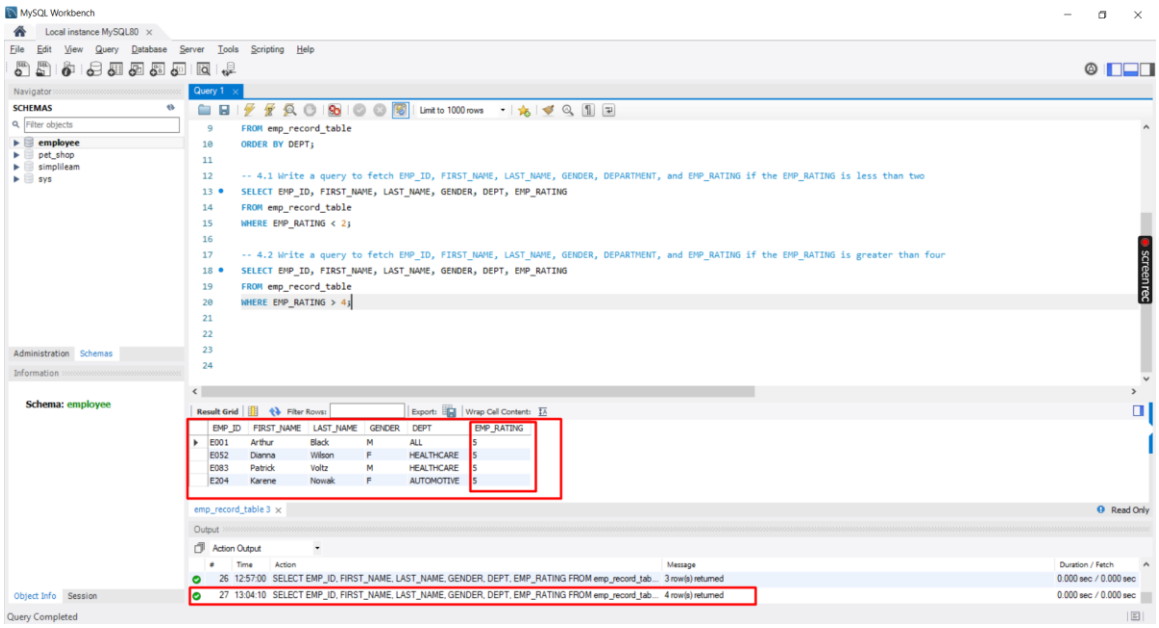


4.2: If the EMP_RATING is greater than four

We can retrieve EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT and EMP_RATING whose EMP_RATING is greater than four from the emp_record_table using the below query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING > 4;
```

Result:

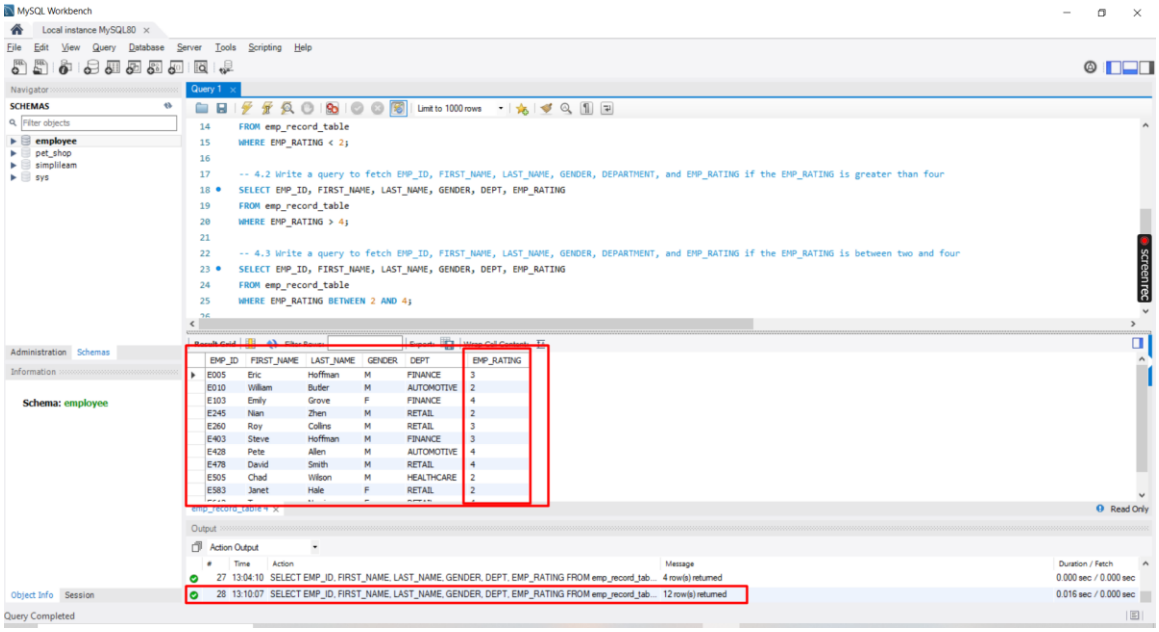


4.3: If the EMP_RATING is between two and four

We can retrieve EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT and EMP_RATING whose EMP_RATING is between two and four from the emp_record_table using the below query:

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING BETWEEN 2 AND 4;
```

Result:

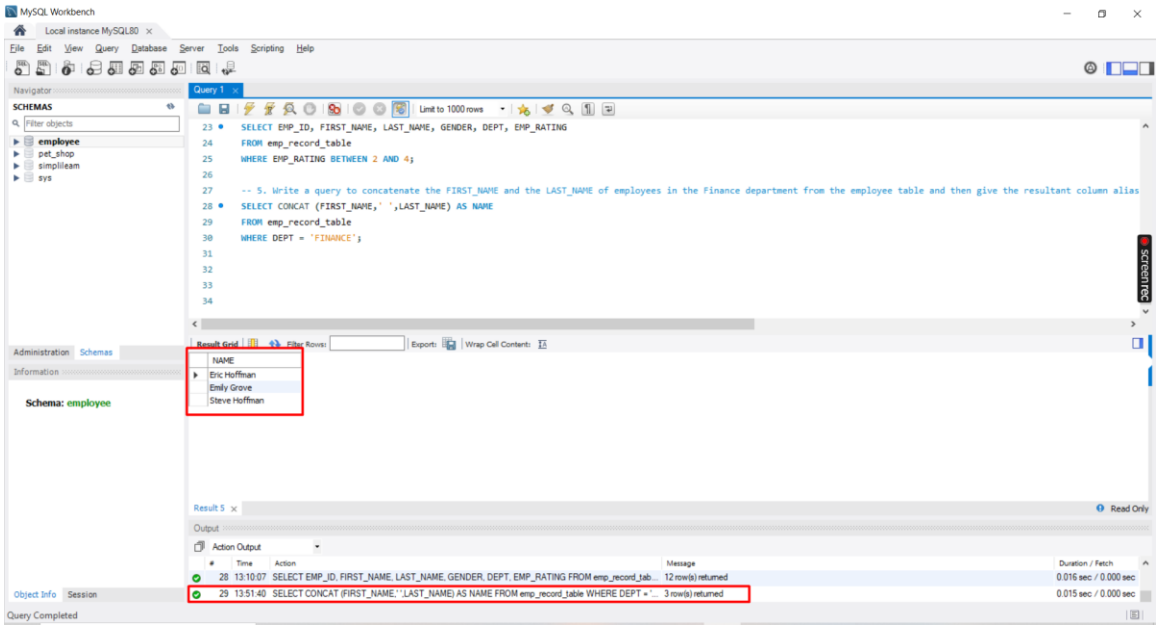


5. Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.

Here we have to concatenate the FIRST_NAME and LAST_NAME from the emp_record_table whose department is Finance and give the name of the table as NAME. We have to execute the below query:

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

Result:



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

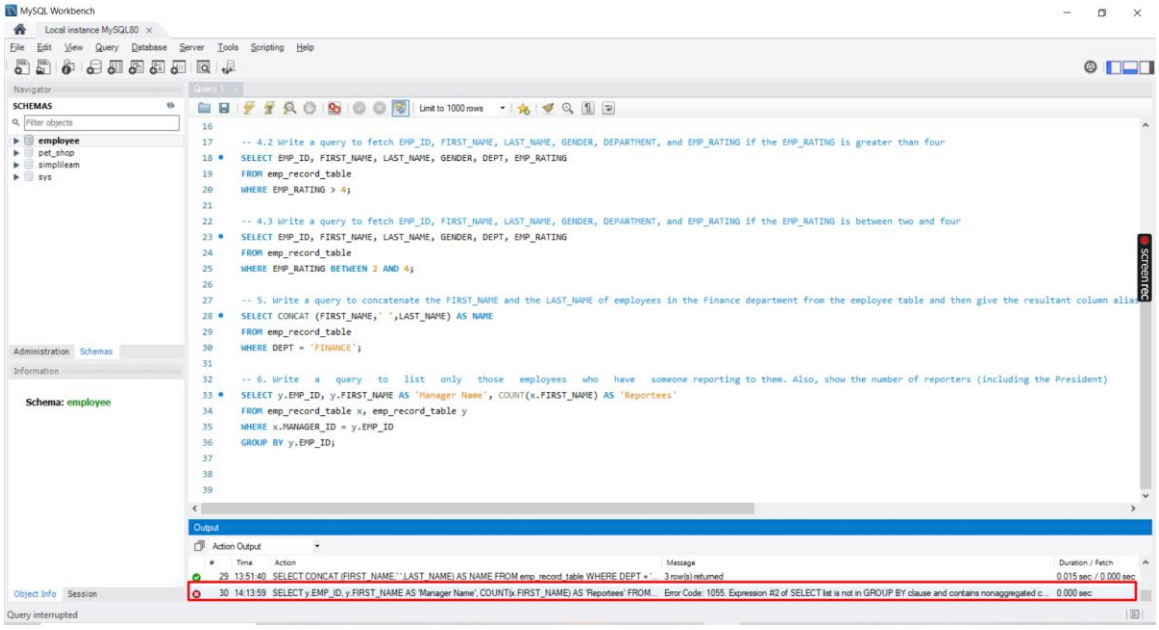
Here we have to retrieve the employees who have someone reporting to them and also showing the number of reportees (including the President).So the query is:

```
SELECT y.EMP_ID,
        y.FIRST_NAME AS 'Manager Name',
        COUNT(x.FIRST_NAME) AS 'Reportees'
FROM emp_record_table x, emp_record_table y
WHERE x.MANAGER_ID = y.EMP_ID
GROUP BY y.EMP_ID;
```

Error Handling:

When the above query is executed we get the error as below:

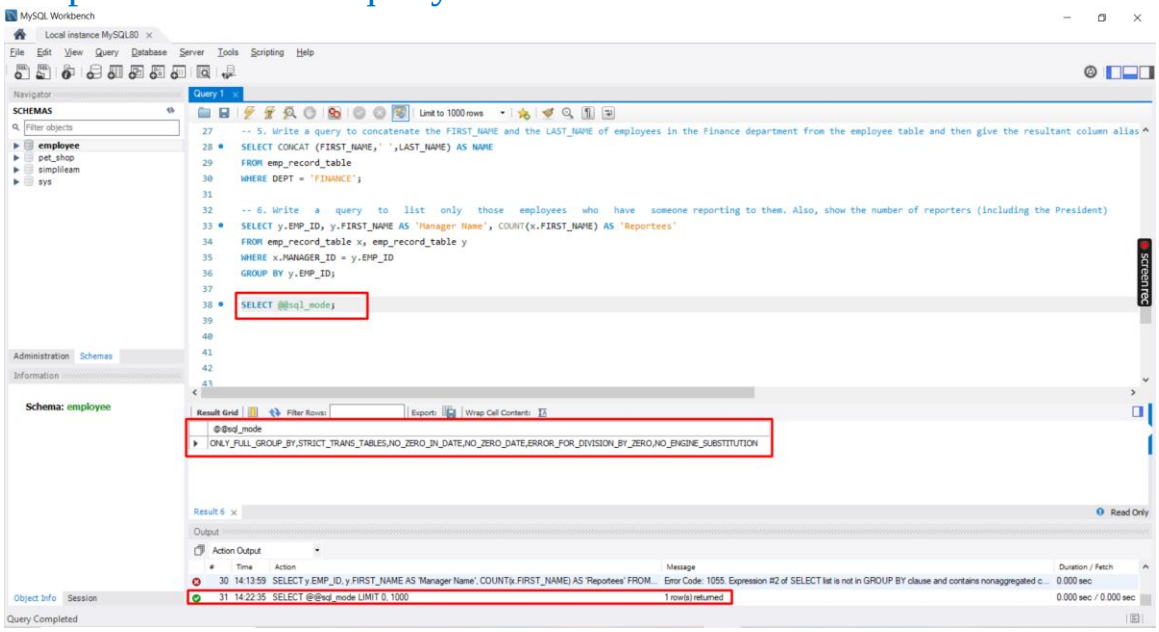
"Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'employee.y.FIRST_NAME' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by 0.000 sec"



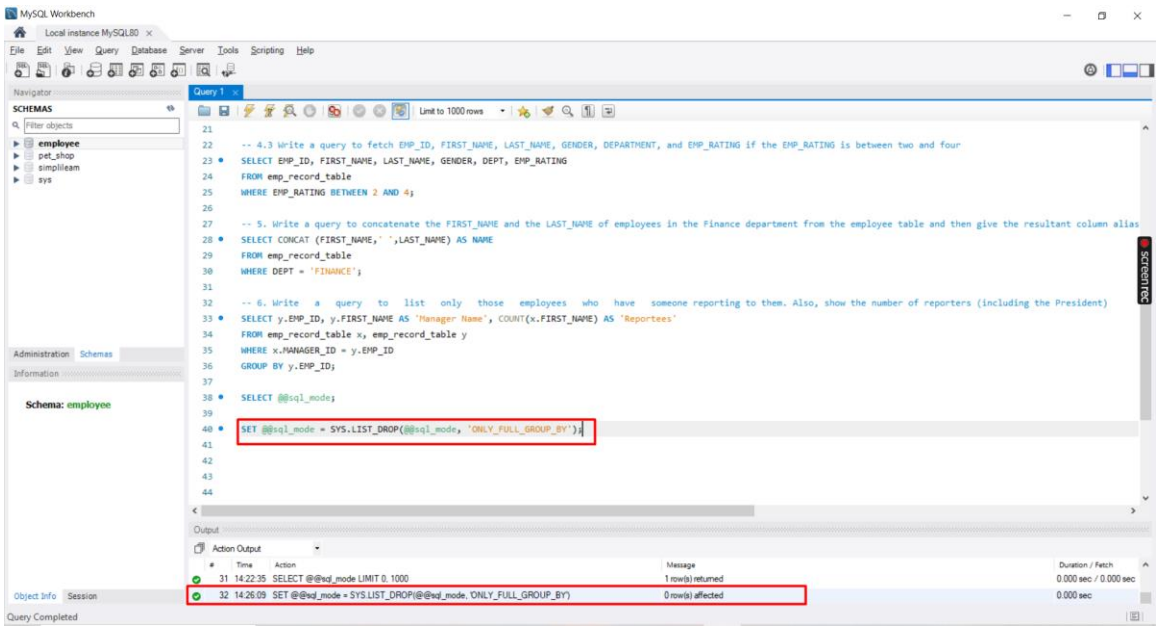
Error code 1055 occurs when we include a column in the SELECT list and we omit it from the GROUP BY clause. This is because our sql_mode contains ONLY_FULL_GROUP_BY enabled.
To check the sql_mode, execute the below query:

```
SELECT @@sql_mode;
```

Output of the above query:



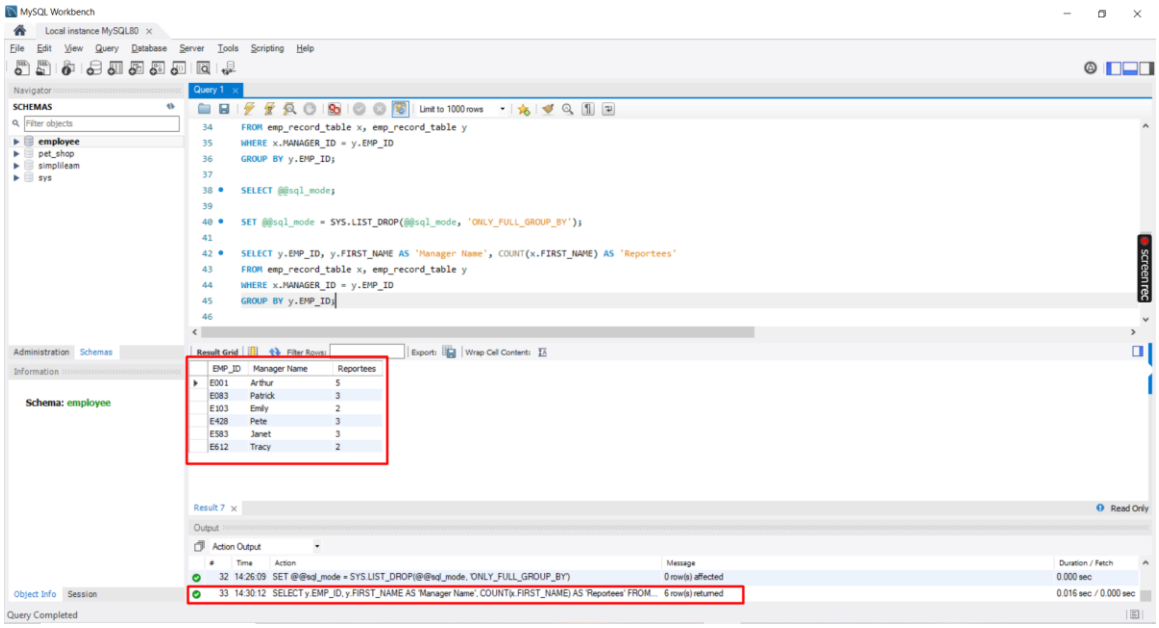
In order to disable the ONLY_FULL_GROUP_BY mode execute the below query:



Now we can execute the below query to see the result of the employees who have someone reporting to them and also showing the number of reportees (including the President).

```
SELECT y.EMP_ID,
            y.FIRST_NAME AS 'Manager Name',
            COUNT(x.FIRST_NAME) AS 'Reportees'
FROM emp_record_table x, emp_record_table y
WHERE x.MANAGER_ID = y.EMP_ID
GROUP BY y.EMP_ID;
```

Result:

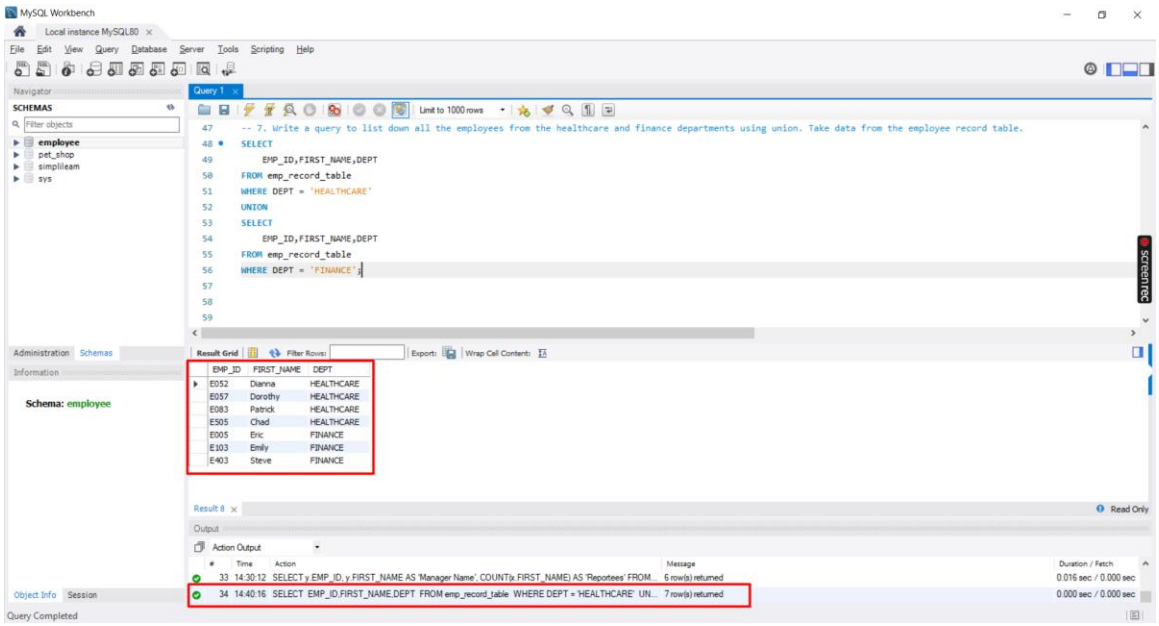


7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.

We have to list all the employees from both healthcare and finance departments using UNION clause. So the query is:

```
SELECT
        EMP_ID,FIRST_NAME,DEPT
FROM emp_record_table
WHERE DEPT = 'HEALTHCARE'
UNION
SELECT
        EMP_ID,FIRST_NAME,DEPT
FROM emp_record_table
WHERE DEPT = 'FINANCE';
```

Result:



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.

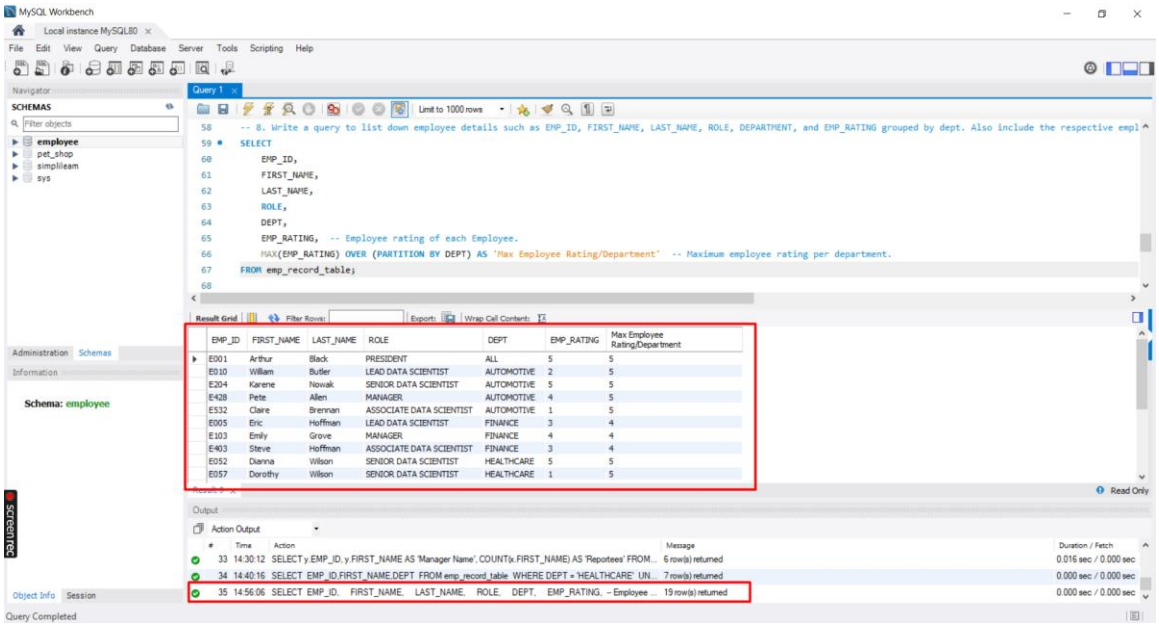
We have to retrieve the employee details such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING from the emp_record_table with the conditions such as

1. Group by department
2. Including the respective employee rating
3. The maximum emp_rating for the department

So the query that should be executed will be:

```
SELECT
        EMP_ID,
        FIRST_NAME,
        LAST_NAME,
        ROLE,
        DEPT,
        EMP_RATING,
        MAX(EMP_RATING) OVER (PARTITION BY DEPT)
                AS 'Max Employee Rating/Department'
FROM emp_record_table;
```

Result:

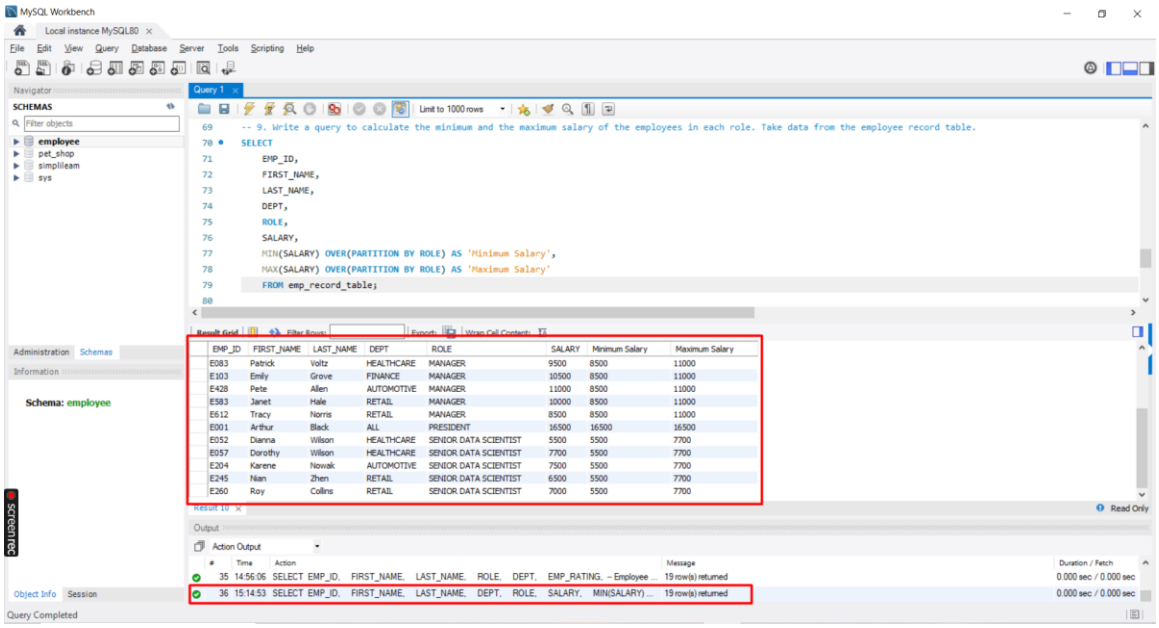


9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

We have to retrieve the EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, and SALARY of each employee, the minimum and maximum salary for that particular role in the company. So the query to be executed will be:

```
SELECT
        EMP_ID,FIRST_NAME,LAST_NAME,
        DEPT,ROLE,SALARY,
        MIN(SALARY) OVER(PARTITION BY ROLE)
            AS 'Minimum Salary',
        MAX(SALARY) OVER(PARTITION BY ROLE)
            AS 'Maximum Salary'
FROM emp_record_table;
```

Result:

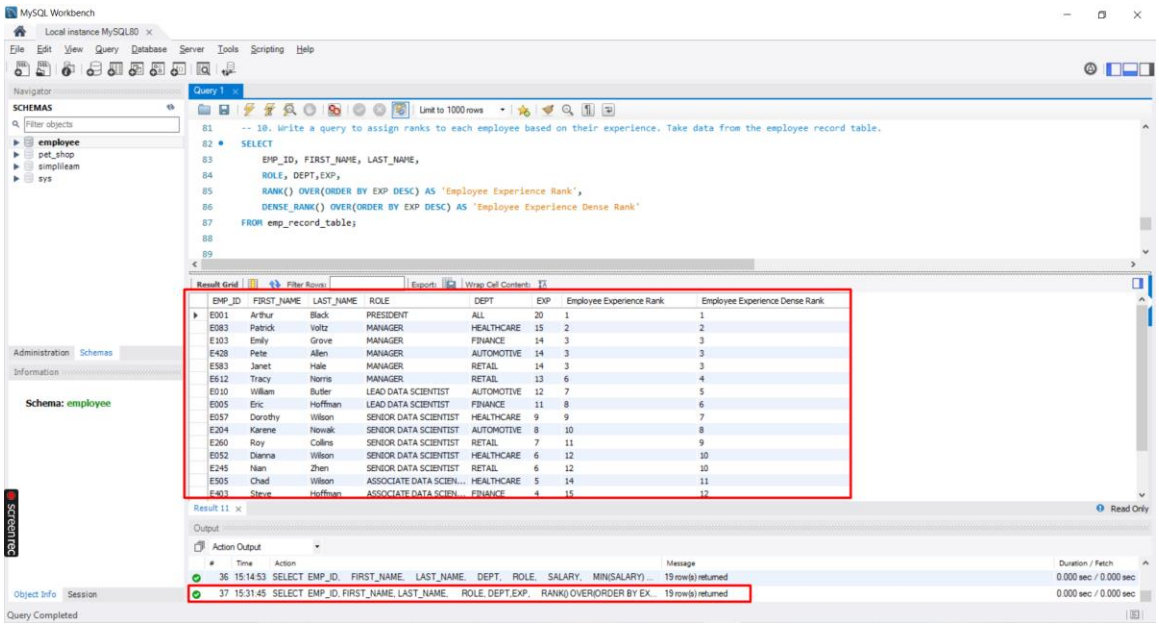


10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

Here we have to assigning ranks to each employee based on their experience and dense rank based on experience. The query to be executed will be:

```
SELECT
        EMP_ID, FIRST_NAME, LAST_NAME,
        ROLE, DEPT,EXP,
        RANK() OVER(ORDER BY EXP DESC)
            AS 'Employee Experience Rank',
        DENSE_RANK() OVER(ORDER BY EXP DESC)
            AS 'Employee Experience Dense Rank'
    FROM emp_record_table;
```

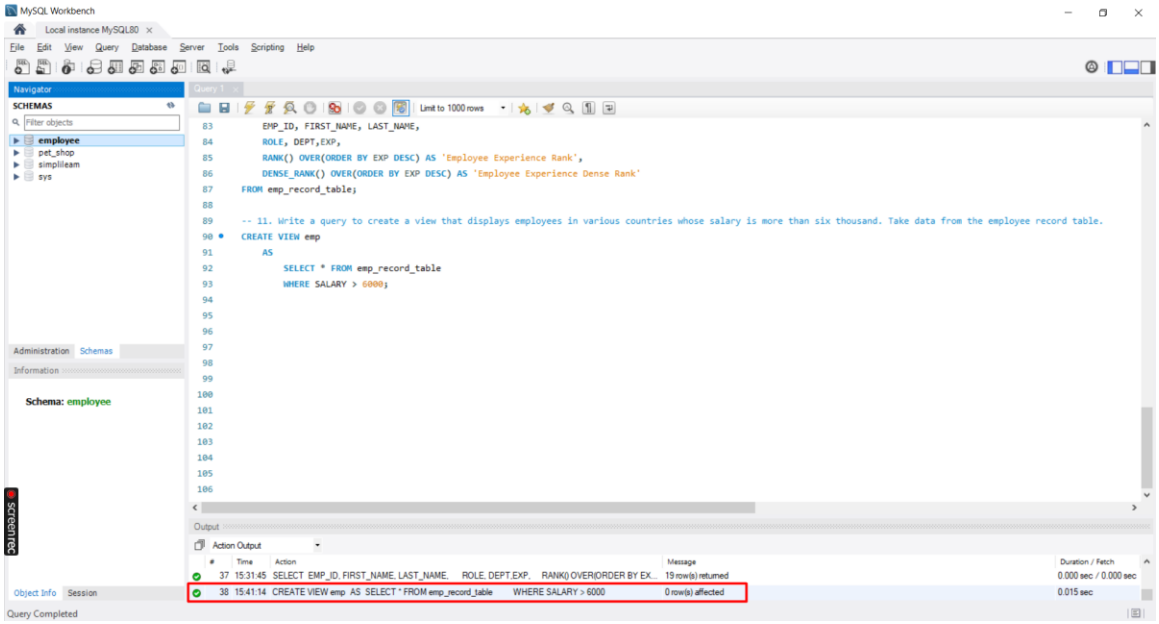
Result:



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.

We have to create a view that displays employees in various countries whose salary is more than six thousand. The query to be executed will be:

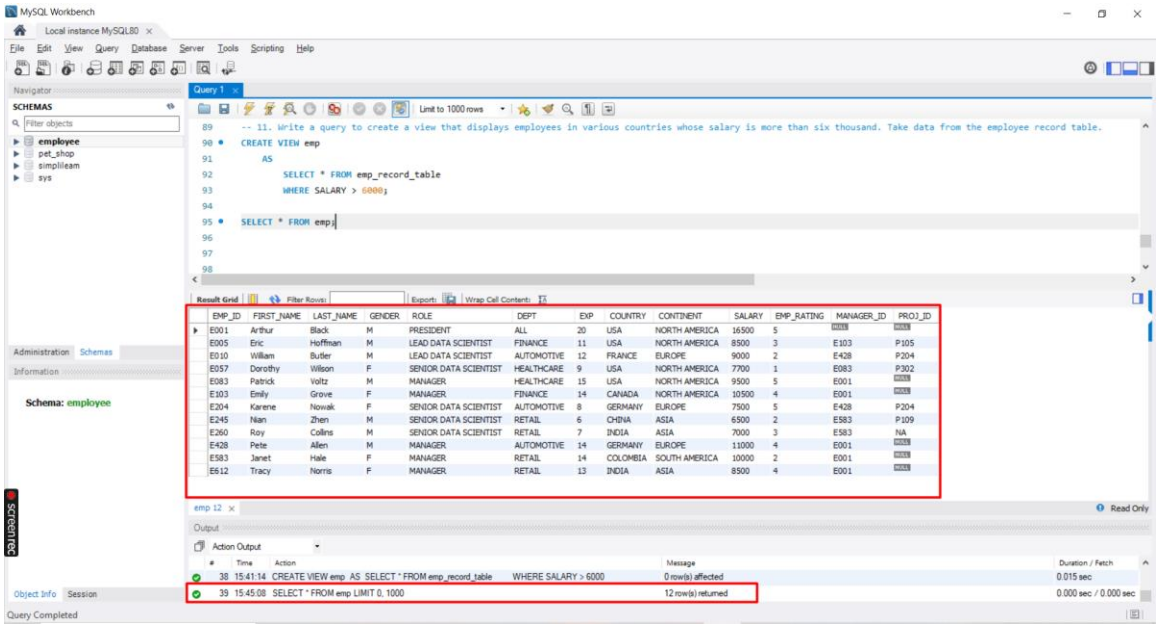
```
CREATE VIEW emp AS
    SELECT * FROM emp_record_table
    WHERE SALARY > 6000;
```



Now we have created the view. Now with the below query we have to list of all employees from various countries whose salary is more than six thousand

```
SELECT * FROM emp;
```

Result:

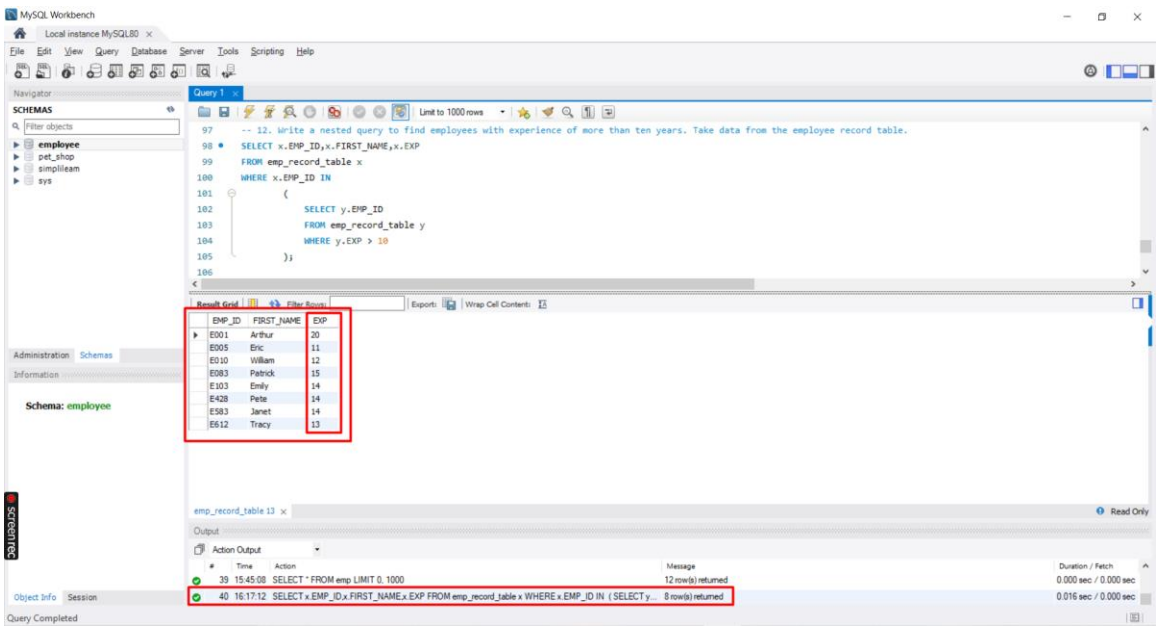

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
employee
pet_shop
simplilearn
sys
Query 1
Limit to 1000 rows
-- 11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.
CREATE VIEW emp
AS
SELECT * FROM emp_record_table
WHERE SALARY > 6000;
SELECT * FROM emp;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
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 16500 5 NULL NULL
E005 Eric Hoffman M LEAD DATA SCIENTIST FINANCE 11 USA NORTH AMERICA 8500 3 E103 P105
E010 William Butler M LEAD DATA SCIENTIST AUTOMOTIVE 12 FRANCE EUROPE 9000 2 E428 P204
E057 Dorothy Wilson F SENIOR DATA SCIENTIST HEALTHCARE 9 USA NORTH AMERICA 7700 1 E083 P302
E083 Patrick Voltz M MANAGER HEALTHCARE 15 USA NORTH AMERICA 9500 5 E001 NULL
E103 Emily Grove F MANAGER FINANCE 14 CANADA NORTH AMERICA 10500 4 E001 NULL
E204 Karene Nowak F SENIOR DATA SCIENTIST AUTOMOTIVE 8 GERMANY EUROPE 7500 5 E428 P204
E245 Nian Zhen M SENIOR DATA SCIENTIST RETAIL 6 CHINA ASIA 6500 2 E583 P109
E260 Roy Collins M SENIOR DATA SCIENTIST RETAIL 7 INDIA ASIA 7000 3 E583 NA
E428 Pete Allen M MANAGER AUTOMOTIVE 14 GERMANY EUROPE 11000 4 E001 NULL
E583 Janet Hale F MANAGER RETAIL 14 COLOMBIA SOUTH AMERICA 10000 2 E001 NULL
E612 Tracy Norris F MANAGER RETAIL 13 INDIA ASIA 8500 4 E001 NULL
Administration Schemas
Information
Schema: employee
emp 12
Read Only
Output
Action Output
Time Action Message Duration / Fetch
38 15:41:14 CREATE VIEW emp AS SELECT * FROM emp_record_table WHERE SALARY > 6000 0 row(s) affected 0.015 sec
39 15:45:08 SELECT * FROM emp LIMIT 0, 1000 12 row(s) returned 0.000 sec / 0.000 sec
Object Info Session
Query Completed

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

We have to retrieve the employees with experience of more than ten years using nested query. The query to be executed will be:

```
SELECT x.EMP_ID,x.FIRST_NAME,x.EXP

FROM emp_record_table x

WHERE x.EMP_ID IN

        (

                SELECT y.EMP_ID

                FROM emp_record_table y

                WHERE y.EXP > 10

        );
```

Result:

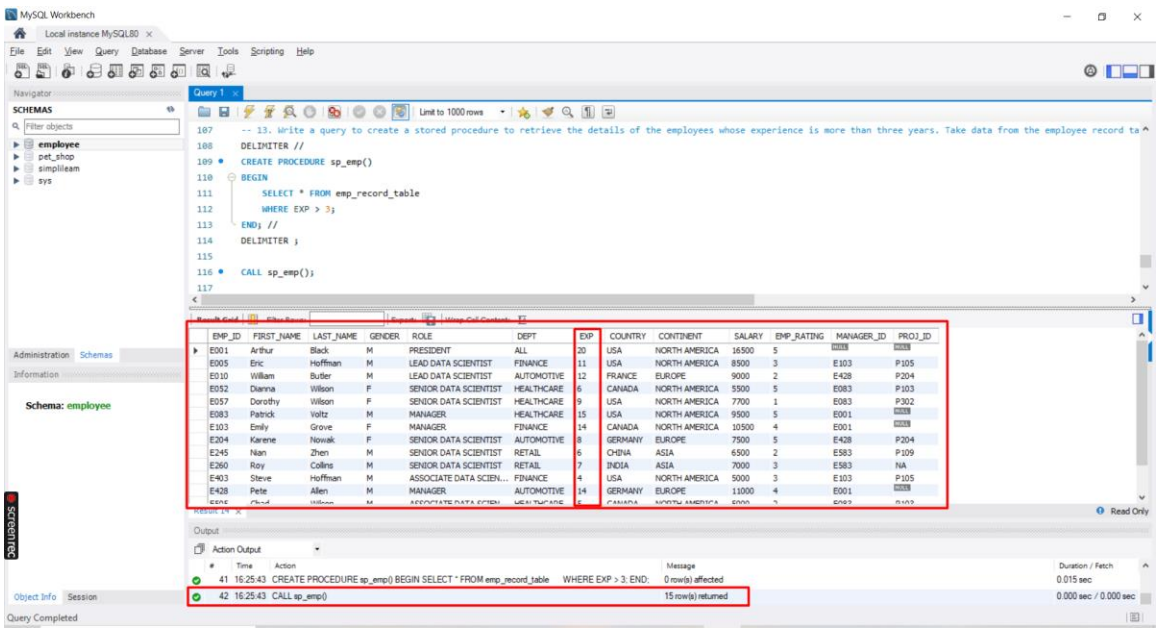


13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

Here we have to create a stored procedure to retrieve the details of the employees whose experience is more than three years. The query that is to be executed is:

```
DELIMITER //
CREATE PROCEDURE sp_emp()
BEGIN
    SELECT * FROM emp_record_table
    WHERE EXP > 3;
END; //
DELIMITER ;
CALL sp_emp();
```

Result:



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'.

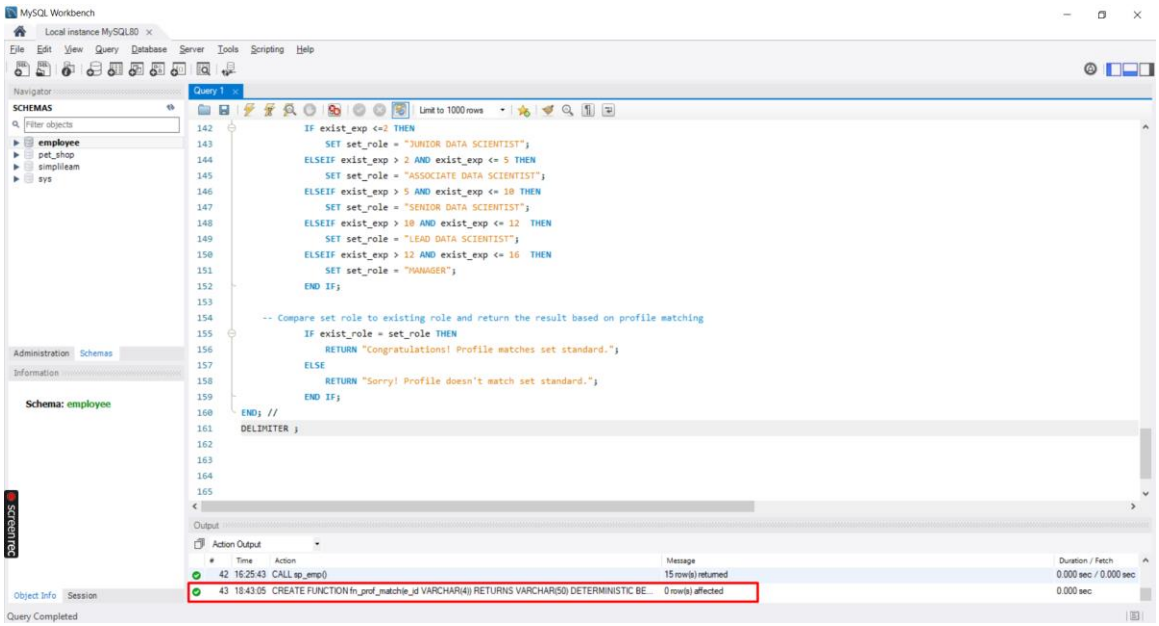
We need to check whether the job profile assigned to each employee in the data_science_team matches the organization's set standard y sing stored functions in the project_table. In order to check the job profiles, the following steps must be performed

- Declaration of variables such as exist_exp, exist_role, and set_role
- Selecting role and experience from data_science_table for the given employee ID
- Evaluating the existing experience and assigning role as per organization's set standard
- Compare set role to existing role and return the result based on profile matching

The query to be executed will be:

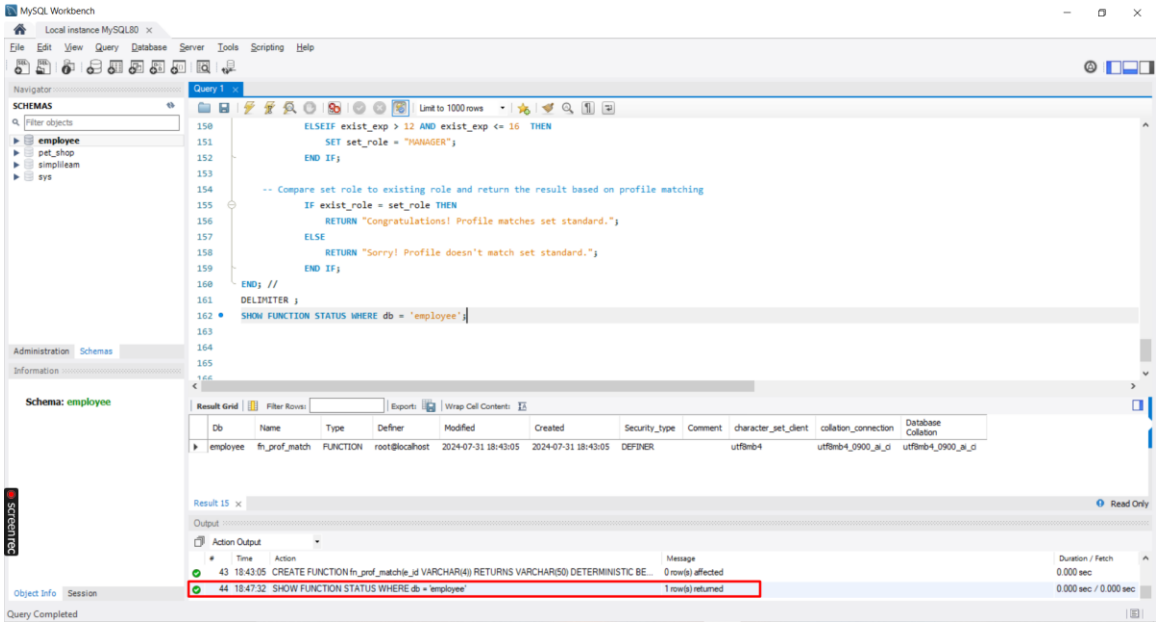
```
DELIMITER //

CREATE FUNCTION fn_prof_match(e_id VARCHAR(4)) RETURNS VARCHAR(50)
DETERMINISTIC

BEGIN


        -- Declaration of variables

                DECLARE exist_exp INT DEFAULT NULL;

                DECLARE exist_role VARCHAR(24) DEFAULT NULL;

                DECLARE set_role VARCHAR(24) DEFAULT NULL;


        -- Selecting experience and role from data_science_team table for the given
employee id.

                SELECT EXP, ROLE INTO exist_exp, exist_role

                FROM data_science_team

                WHERE EMP_ID = e_id;


        -- Evaluating the existing experience and assigning role as per organization's set
standard

                IF exist_exp <=2 THEN

                        SET set_role = "JUNIOR DATA SCIENTIST";

                ELSEIF exist_exp > 2 AND exist_exp <= 5 THEN

                        SET set_role = "ASSOCIATE DATA SCIENTIST";

                ELSEIF exist_exp > 5 AND exist_exp <= 10 THEN

                        SET set_role = "SENIOR DATA SCIENTIST";

                ELSEIF exist_exp > 10 AND exist_exp <= 12  THEN
```

```
            SET set_role = "LEAD DATA SCIENTIST";
        ELSEIF exist_exp > 12 AND exist_exp <= 16  THEN
            SET set_role = "MANAGER";
        END IF;

    -- Compare set role to existing role and return the result based on profile matching
        IF exist_role = set_role THEN
            RETURN "Congratulations! Profile matches set standard.";
        ELSE
            RETURN "Sorry! Profile doesn't match set standard.";
        END IF;
END; //
DELIMITER;
```



Once the stored function is created as shown in the picture above, use the below query

```
SHOW FUNCTION STATUS WHERE db = 'employee';
```

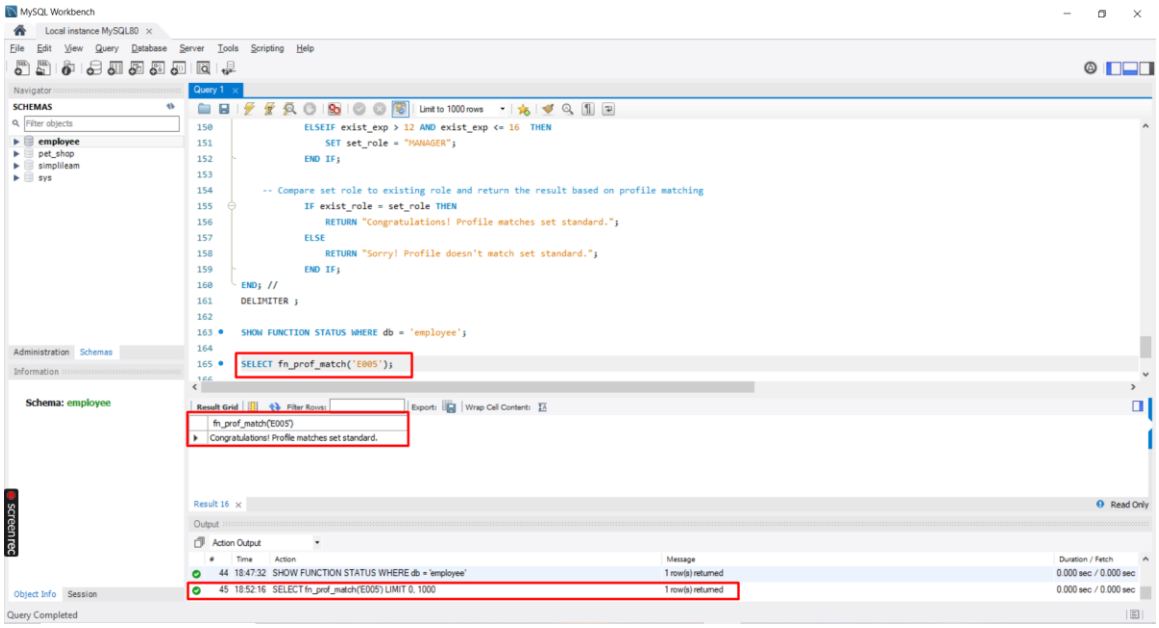


To verify the roles and also to obtain the predefined messages as outcome, execute the below query

Mention the EMP_ID available in the database

```
SHOW FUNCTION STATUS WHERE db = 'employee';
```

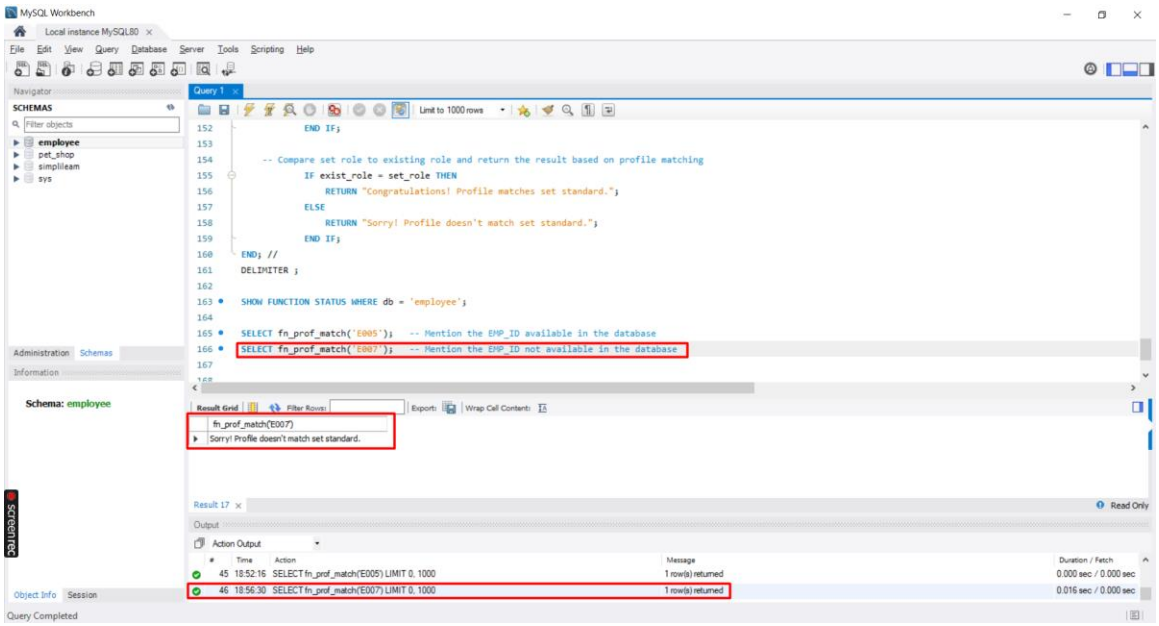
Result:



Mention the EMP_ID not available in the database

```
SELECT fn_prof_match('E007');
```

Result:



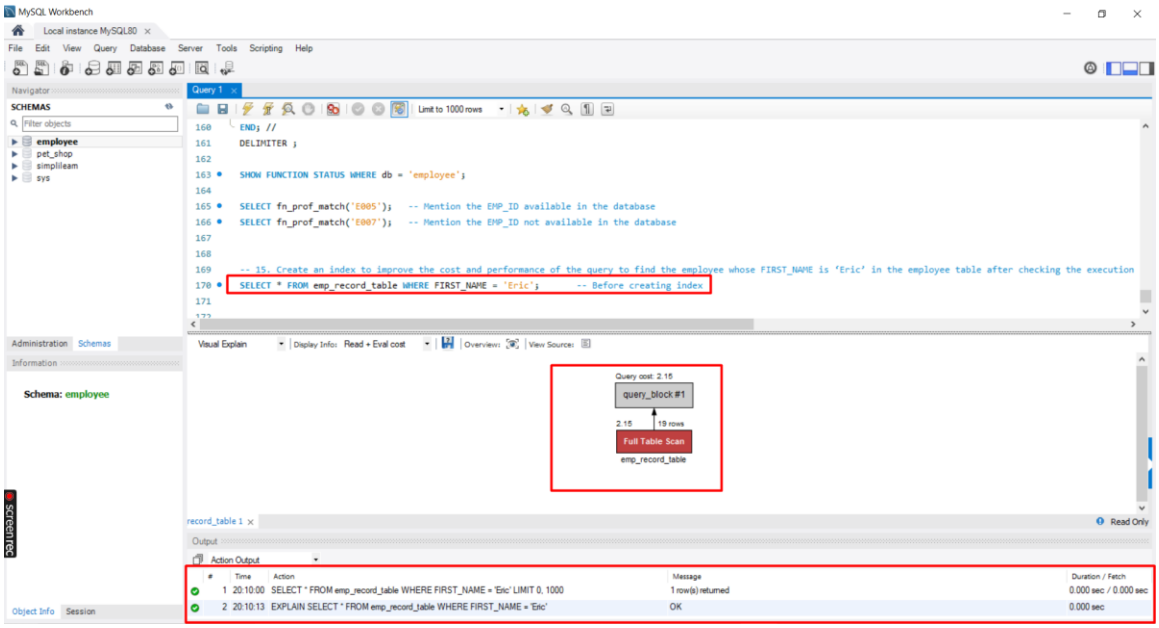
15. Create an index to improve the cost and performance of the query to find the employee whose FIRST_NAME is ‘Eric’ in the employee table after checking the execution plan

We have to 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 by comparing the execution plan before and after creating the index. The query to be executed is:

- Before creating index:

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

Result:



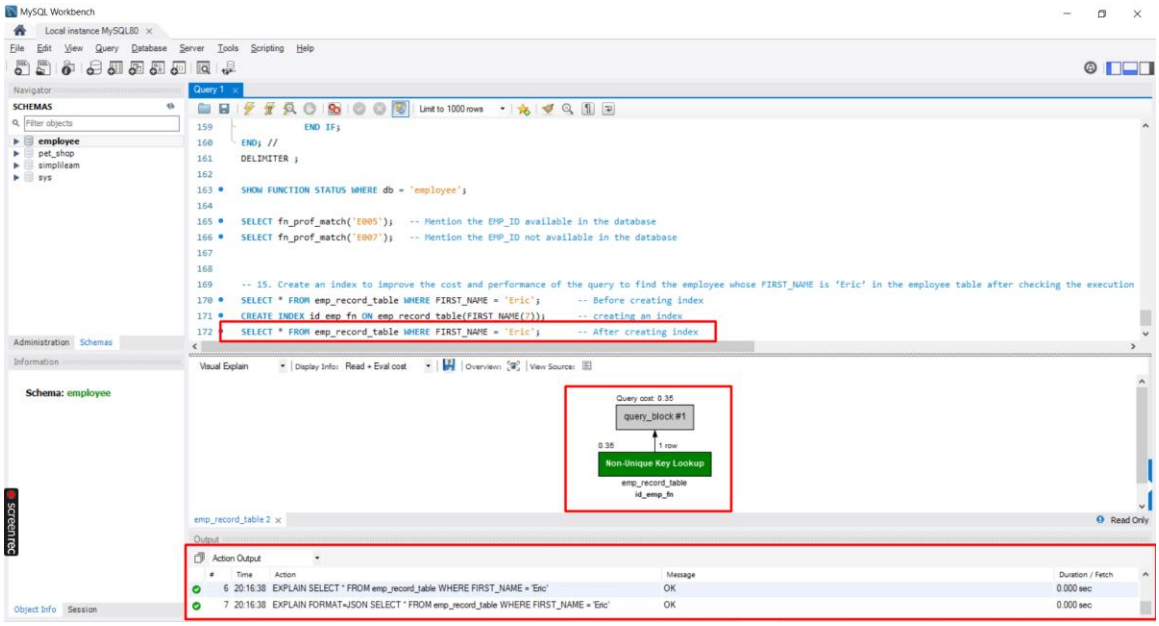
Now we are creating an index using the below query:

```
CREATE INDEX id_emp_fn ON emp_record_table(FIRST_NAME(7));
```

- After creating index:

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

Result:

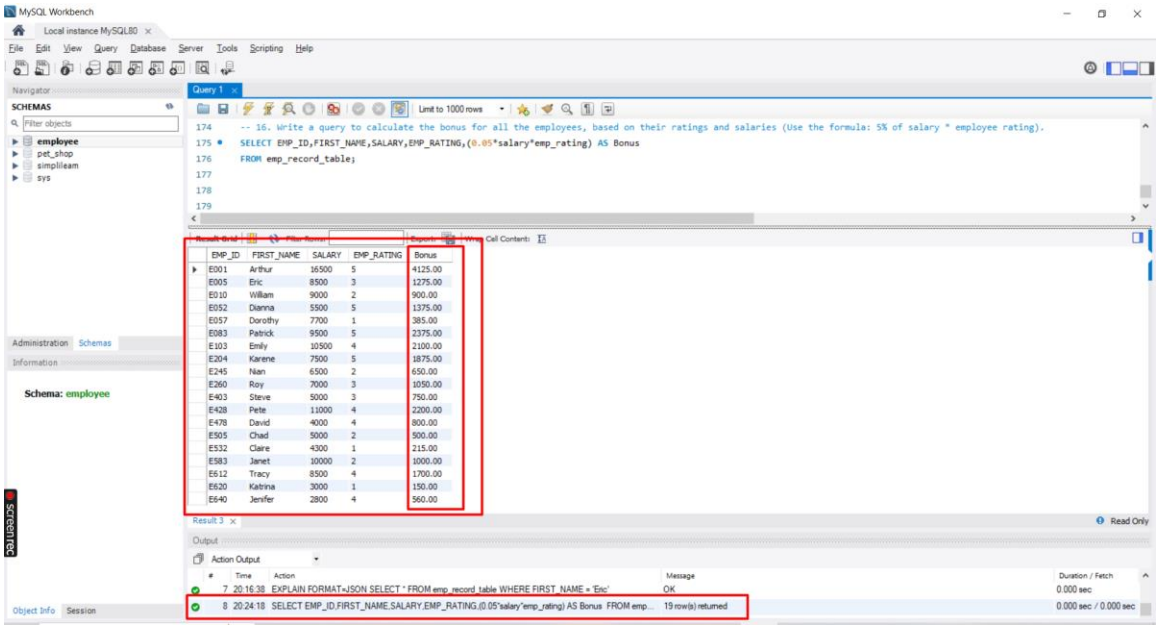

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
employee
pet_shop
simplilearn
sys
Query 1
Limit to 1000 rows
END IF;
END; //
DELIMITER ;
SHOW FUNCTION STATUS WHERE db = 'employee';
SELECT fn_prof_match('E005'); -- Mention the EMP_ID available in the database
SELECT fn_prof_match('E007'); -- Mention the EMP_ID not available in the database
-- 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
SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric'; -- Before creating index
CREATE INDEX id_emp_fn ON emp_record_table(FIRST_NAME(7)); -- creating an index
SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric'; -- After creating index
Administration Schemas
Information
Schema: employee
Visual Explain
Display Info: Read + Eval cost
Overview
View Source
Query cost: 0.35
query_block #1
0.35
1 row
Non-Unique Key Lookup
emp_record_table
id_emp_fn
emp_record_table 2
Read Only
Output
Action Output
Time Action Message Duration / Fetch
6 20:16:38 EXPLAIN SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric' OK 0.000 sec
7 20:16:38 EXPLAIN FORMAT=JSON SELECT * FROM emp_record_table WHERE FIRST_NAME = 'Eric' OK 0.000 sec
Object Info Session

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).

We have to calculate the bonus for all the employees based on their ratings and salaries. The query that is to be executed will be:

```
SELECT

EMP_ID,FIRST_NAME,SALARY,EMP_RATING,(0.05*salary*emp_rating) AS Bonus

FROM emp_record_table;
```

Result:



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

Here we will calculate the average salary distribution for each employee based on the continent and country. The query that is to be executed is:

```
SELECT
        EMP_ID,FIRST_NAME,SALARY,COUNTRY,CONTINENT,
        AVG(SALARY) OVER(PARTITION BY COUNTRY)
                AS 'Average Country Based Salary',
        AVG(SALARY) OVER(PARTITION BY CONTINENT)
                AS 'Average Continent Based Salary'
FROM emp_record_table;
```

Result: