

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.

Sol:- CREATE DATABASE IF NOT EXISTS employee;

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

DESCRIBE emp_record_table;

```
3 • DESCRIBE emp_record_table;
4 • DESCRIBE Proj_table;
5 • DESCRIBE Data_science_team;
6
7 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
8 FROM emp_record_table;
```

Result Grid Filter Rows: Export: Wrap Cell Content:						
Field	Type	Null	Key	Default	Extra	
EMP_ID	text	YES		NULL		
FIRST_NAME	text	YES		NULL		
LAST_NAME	text	YES		NULL		
GENDER	text	YES		NULL		
ROLE	text	YES		NULL		
DEPT	text	YES		NULL		

DESCRIBE Proj_table;

```
4 • DESCRIBE Proj_table;
5 • DESCRIBE Data_science_team;
6
7 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
8 FROM emp_record_table;
```

Result Grid Filter Rows: Export: Wrap Cell Content:						
Field	Type	Null	Key	Default	Extra	
PROJECT_ID	text	YES		NULL		
PROJ_NAME	text	YES		NULL		
DOMAIN	text	YES		NULL		
START_DATE	text	YES		NULL		
CLOSURE_DATE	text	YES		NULL		
DEV_QTR	text	YES		NULL		

DESCRIBE Data_science_team;

```
5 • DESCRIBE Data_science_team;
6
7 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
8 FROM emp_record_table;
```

Result Grid Filter Rows: Export: Wrap Cell Content:						
Field	Type	Null	Key	Default	Extra	
EMP_ID	text	YES		NULL		
FIRST_NAME	text	YES		NULL		
LAST_NAME	text	YES		NULL		
GENDER	text	YES		NULL		
ROLE	text	YES		NULL		
DEPT	text	YES		NULL		

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;

```

7 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT
8   FROM emp_record_table;
9
10 • ALTER TABLE emp_record_table
11   ADD COLUMN DEPARTMENT VARCHAR(50);

```

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

ALTER TABLE emp_record_table

ADD COLUMN DEPARTMENT VARCHAR(50);

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

(a) less than two

(b) greater than four

(c) between two and four

(a) SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING < 2;

```

13 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
14   FROM emp_record_table
15   WHERE EMP_RATING < 2;
16

```

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPARTMENT	EMP_RATING
E057	Dorothy	Wilson	F	NULL	1
E532	Claire	Brennan	F	NULL	1
E620	Katrina	Allen	F	NULL	1

(b) SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING > 4;

```

17 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
18 FROM emp_record_table
19 WHERE EMP_RATING > 4;
20
21 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
22 FROM emp_record_table

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPARTMENT	EMP_RATING
E001	Arthur	Black	M	NULL	5
E052	Dianna	Wilson	F	NULL	5
E083	Patrick	Voltz	M	NULL	5
E204	Karene	Nowak	F	NULL	5

Form Editor

(c) SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
FROM emp_record_table
WHERE EMP_RATING BETWEEN 2 AND 4;

```

21 • SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATING
22 FROM emp_record_table
23 WHERE EMP_RATING BETWEEN 2 AND 4;
24
25 • SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPARTMENT	EMP_RATING
E005	Eric	Hoffman	M	NULL	3
E010	William	Butler	M	NULL	2
E103	Emily	Grove	F	NULL	4
E245	Nian	Zhen	M	NULL	2
E260	Roy	Collins	M	NULL	3
E403	Steve	Hoffman	M	NULL	3

emp_record_table 30 x

Read Only

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 DEPARTMENT = 'Finance';

```

25 • SELECT CONCAT(FIRST_NAME, ' ', LAST_NAME) AS NAME
26 FROM emp_record_table
27 WHERE DEPARTMENT = 'Finance';
28
29 • SELECT

```

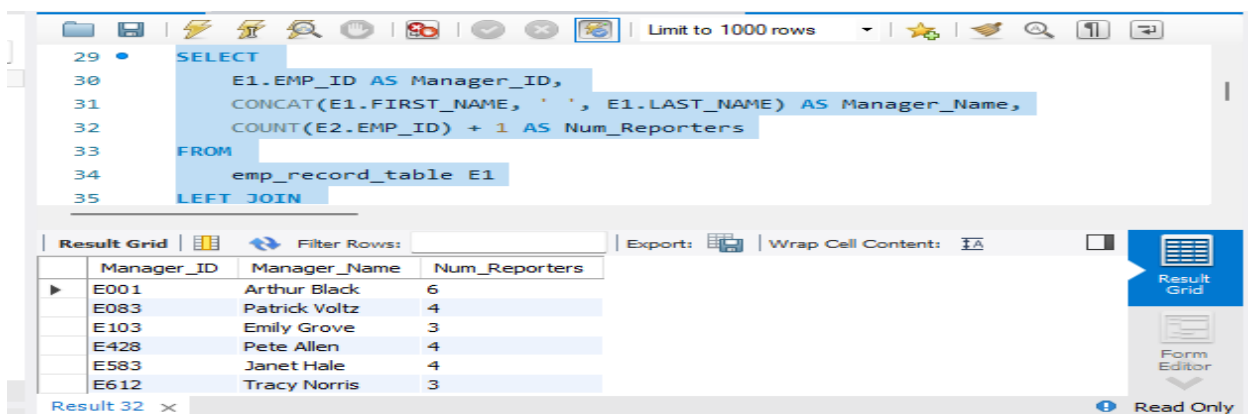
Result Grid

NAME

Form Editor

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
E1.EMP_ID AS Manager_ID,
CONCAT(E1.FIRST_NAME, ' ', E1.LAST_NAME) AS Manager_Name,
COUNT(E2.EMP_ID) + 1 AS Num_Reporters
FROM
emp_record_table E1
LEFT JOIN
emp_record_table E2 ON E1.EMP_ID = E2.MANAGER_ID
GROUP BY
E1.EMP_ID, E1.FIRST_NAME, E1.LAST_NAME
HAVING
Num_Reporters > 1;
```



The screenshot shows a database query editor with the following SQL query:

```
29 SELECT
30 E1.EMP_ID AS Manager_ID,
31 CONCAT(E1.FIRST_NAME, ' ', E1.LAST_NAME) AS Manager_Name,
32 COUNT(E2.EMP_ID) + 1 AS Num_Reporters
33 FROM
34 emp_record_table E1
35 LEFT JOIN
```

Below the query editor, the results are displayed in a grid. The grid has four columns: Manager_ID, Manager_Name, and Num_Reporters. The results are as follows:

Manager_ID	Manager_Name	Num_Reporters
E001	Arthur Black	6
E083	Patrick Voltz	4
E103	Emily Grove	3
E428	Pete Allen	4
E583	Janet Hale	4
E612	Tracy Norris	3

The interface also includes a toolbar at the top with various icons, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button on the right. The status bar at the bottom indicates 'Result 32' and 'Read Only'.

The screenshot shows a SQL query editor with the following query:

```
35 LEFT JOIN
36 emp_record_table E2 ON E1.EMP_ID = E2.MANAGER_ID
37 GROUP BY
38 E1.EMP_ID, E1.FIRST_NAME, E1.LAST_NAME
39 HAVING
40 Num_Reporters > 1;
41
```

Below the query editor is a 'Result Grid' showing the results of the query:

Manager_ID	Manager_Name	Num_Reporters
E001	Arthur Black	6
E083	Patrick Voltz	4
E103	Emily Grove	3
E428	Pete Allen	4
E583	Janet Hale	4
E612	Tracy Norris	3

The interface includes a toolbar at the top with icons for file operations, a 'Limit to 1000 rows' dropdown, and buttons for 'Result Grid', 'Form Editor', and 'Read Only'.

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, DEPARTMENT
FROM emp_record_table
WHERE DEPARTMENT = 'Healthcare'
UNION
SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT
FROM emp_record_table
WHERE DEPARTMENT = 'Finance';
```

The screenshot shows a SQL query editor with the following query:

```
43 SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT
44 FROM emp_record_table
45 WHERE DEPARTMENT = 'Healthcare'
46 UNION
47 SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT
48 FROM emp_record_table
49 WHERE DEPARTMENT = 'Finance';
```

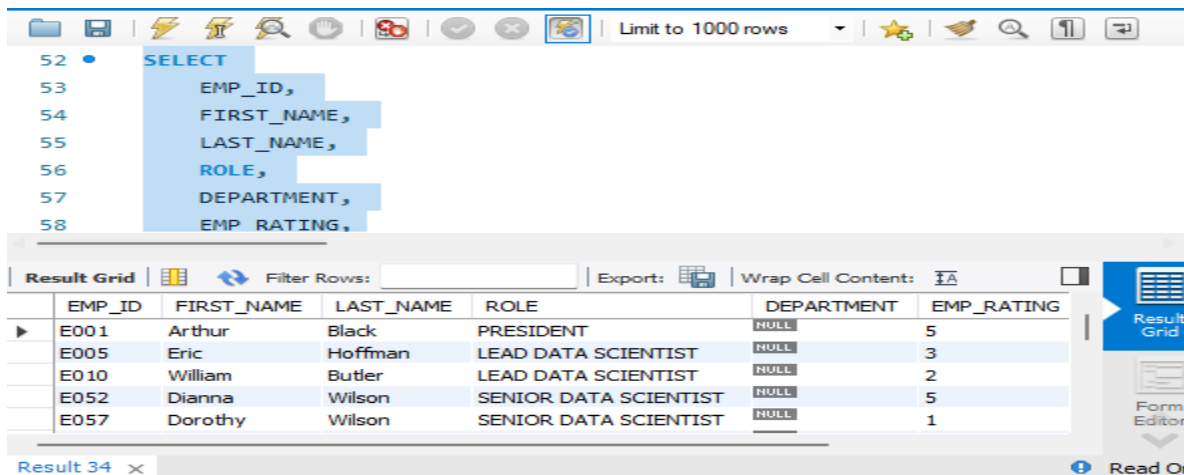
Below the query editor is a 'Result Grid' showing the results of the query:

EMP_ID	FIRST_NAME	LAST_NAME	DEPARTMENT
--------	------------	-----------	------------

The interface includes a toolbar at the top with icons for file operations, a 'Limit to 1000 rows' dropdown, and buttons for 'Result Grid', 'Form Editor', and 'Read Only'.

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  
EMP_ID,  
FIRST_NAME,  
LAST_NAME,  
ROLE,  
DEPARTMENT,  
EMP_RATING,  
MAX(EMP_RATING) OVER(PARTITION BY DEPARTMENT) AS Max_Rating_For_Dept  
FROM  
emp_record_table;
```



The screenshot shows a database query editor with a toolbar at the top. The SQL query is entered in the editor area, and the results are displayed in a 'Result Grid' below. The query is:
`SELECT
EMP_ID,
FIRST_NAME,
LAST_NAME,
ROLE,
DEPARTMENT,
EMP_RATING,
MAX(EMP_RATING) OVER(PARTITION BY DEPARTMENT) AS Max_Rating_For_Dept
FROM
emp_record_table;`
The 'Result Grid' shows the following data:

EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPARTMENT	EMP_RATING
E001	Arthur	Black	PRESIDENT	NULL	5
E005	Eric	Hoffman	LEAD DATA SCIENTIST	NULL	3
E010	William	Butler	LEAD DATA SCIENTIST	NULL	2
E052	Dianna	Wilson	SENIOR DATA SCIENTIST	NULL	5
E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	NULL	1

SQL File 9* SCIENCE_TECH_PROJECT_S... data_science_team emp_record_table proj_

Limit to 1000 rows

```

57 DEPARTMENT,
58 EMP_RATING,
59 MAX(EMP_RATING) OVER(PARTITION BY DEPARTMENT) AS Max_Rating_For_Dept
60 FROM
61 emp_record_table;
62
63

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPARTMENT	EMP_RATING
▶	E001	Arthur	Black	PRESIDENT	NULL	5
	E005	Eric	Hoffman	LEAD DATA SCIENTIST	NULL	3
	E010	William	Butler	LEAD DATA SCIENTIST	NULL	2
	E052	Dianna	Wilson	SENIOR DATA SCIENTIST	NULL	5
	E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	NULL	1

Result 34

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 Min_Salary,
MAX(SALARY) AS Max_Salary
FROM
emp_record_table
GROUP BY
ROLE;

```

SQL File 9* SCIENCE_TECH_PROJECT_S... data_science_team emp_record_table proj_

Limit to 1000 rows

```

64 SELECT
65 ROLE,
66 MIN(SALARY) AS Min_Salary,
67 MAX(SALARY) AS Max_Salary
68 FROM
69 emp_record_table
70 GROUP BY

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	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

Result 35

SQL File 9

SCIENCE TECH PROJECT S

data_science_team

emp_record_table

proj_

Limit to 1000 rows

67 MAX(SALARY) AS Max_Salary

68 FROM

69 emp_record_table

70 GROUP BY

71 ROLE;

72

73

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result Grid

Form Editor

Read Only

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

Result 35 x

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,
RANK() OVER (ORDER BY EXP DESC) AS Experience_Rank
FROM
emp_record_table;

```

SQL File 9*

SCIENCE TECH PROJECT S

data_science_team

emp_record_table

proj_

Limit to 1000 rows

74 SELECT

75 EMP_ID,

76 FIRST_NAME,

77 LAST_NAME,

78 EXP,

79 RANK() OVER (ORDER BY EXP DESC) AS Experience_Rank

80 FROM

emp_record_table;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Result Grid

Form Editor

Read Only

EMP_ID	FIRST_NAME	LAST_NAME	EXP	Experience_Rank
E001	Arthur	Black	20	1
E083	Patrick	Voltz	15	2
E103	Emily	Grove	14	3
E428	Pete	Allen	14	3
E583	Janet	Hale	14	3
E612	Tracy	Norris	13	6

Result 36 x

Output

Action Output

#	Time	Action	Message
88	01:48:34	SELECT CONCAT(FIRST_NAME, '', LAST_NAME) AS NAME FROM emp_record_table W...	0 row(s) returned
89	01:49:25	SELECT E1.EMP_ID AS Manager_ID, CONCAT(E1.FIRST_NAME, '', E1.LAST_NA...	6 row(s) returned
90	01:50:39	SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT FROM emp_record_table ...	0 row(s) returned
91	01:52:43	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, ...	19 row(s) returned
92	01:53:51	SELECT ROLE, MIN(SALARY) AS Min_Salary, MAX(SALARY) AS Max_Salary FR...	6 row(s) returned
93	01:55:25	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, RANK() OVER (ORD...	19 row(s) returned

SQL File 9* SCIENCE_TECH_PROJECT_S data_science_team emp_record_table proj_

77 LAST_NAME,
 78 EXP,
 79 RANK() OVER (ORDER BY EXP DESC) AS Experience_Rank
 80 FROM
 81 emp_record_table;
 82
 83

Result Grid Filter Rows: Export: Wrap Cell Content:

EMP_ID	FIRST_NAME	LAST_NAME	EXP	Experience_Rank
E001	Arthur	Black	20	1
E083	Patrick	Voltz	15	2
E103	Emily	Grove	14	3
E428	Pete	Allen	14	3
E583	Janet	Hale	14	3
E612	Tracy	Norris	13	6

Result 36 x Read Only

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 high_salary_employees AS
SELECT
  EMP_ID,
  FIRST_NAME,
  LAST_NAME,
  SALARY,
  COUNTRY
FROM
  emp_record_table
WHERE
  SALARY > 6000;
SELECT * FROM high_salary_employees;
```

84
85 • **CREATE VIEW** high_salary_employees **AS**
86 **SELECT**
87 EMP_ID,
88 FIRST_NAME,
89 LAST_NAME,
90 SALARY,

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | **Result Grid** | **Form Editor**

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY
E001	Arthur	Black	16500	USA
E005	Eric	Hoffman	8500	USA
E010	William	Butler	9000	FRANCE
E057	Dorothy	Wilson	7700	USA
E083	Patrick	Voltz	9500	USA
E103	Emily	Grove	10500	CANADA

high_salary_employees37 x

Output

Action Output

#	Time	Action	Message
90	01:50:39	SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT FROM emp_record_table ...	0 row(s) returned
91	01:52:43	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, ...	19 row(s) returned
92	01:53:51	SELECT ROLE, MIN(SALARY) AS Min_Salary, MAX(SALARY) AS Max_Salary FR...	6 row(s) returned
93	01:55:25	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, RANK() OVER (ORD...	19 row(s) returned
94	01:57:47	CREATE VIEW high_salary_employees AS SELECT EMP_ID, FIRST_NAME, LA...	Error Code: 1050
95	01:57:56	SELECT * FROM high_salary_employees LIMIT 0, 1000	12 row(s) returned

SQL File 9* SCIENCE_TECH_PROJECT_S_ data_science_team emp_record_table proj_ | Limit to 1000 rows

92 **FROM**
93 emp_record_table
94 **WHERE**
95 SALARY > 6000;
96
97
98 • **SELECT * FROM high_salary_employees;**

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | **Result Grid** | **Form Editor**

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY
E001	Arthur	Black	16500	USA
E005	Eric	Hoffman	8500	USA
E010	William	Butler	9000	FRANCE
E057	Dorothy	Wilson	7700	USA
E083	Patrick	Voltz	9500	USA
E103	Emily	Grove	10500	CANADA

high_salary_employees37 x

Output

Action Output

#	Time	Action	Message
90	01:50:39	SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT FROM emp_record_table ...	0 row(s) returned
91	01:52:43	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, ...	19 row(s) returned
92	01:53:51	SELECT ROLE, MIN(SALARY) AS Min_Salary, MAX(SALARY) AS Max_Salary FR...	6 row(s) returned
93	01:55:25	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, RANK() OVER (ORD...	19 row(s) returned
94	01:57:47	CREATE VIEW high_salary_employees AS SELECT EMP_ID, FIRST_NAME, LA...	Error Code: 1050
95	01:57:56	SELECT * FROM high_salary_employees LIMIT 0, 1000	12 row(s) returned

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 > 10;

The screenshot shows a database management tool interface. The top pane displays a SQL query:

```
SELECT  
EMP_ID,  
FIRST_NAME,  
LAST_NAME,  
EXP  
FROM  
emp_record_table
```

The bottom pane shows the 'Result Grid' with the following data:

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

Below the result grid, the 'Output' pane shows a list of actions and their results:

#	Time	Action	Message
91	01:52:43	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, ...	19 row(s) returned
92	01:53:51	SELECT ROLE, MIN(SALARY) AS Min_Salary, MAX(SALARY) AS Max_Salary FR...	6 row(s) returned
93	01:55:25	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, RANK() OVER (ORD...	19 row(s) returned
94	01:57:47	CREATE VIEW high_salary_employees AS SELECT EMP_ID, FIRST_NAME, LA...	Error Code: 105
95	01:57:56	SELECT * FROM high_salary_employees LIMIT 0, 1000	12 row(s) returned
96	02:00:21	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_t...	8 row(s) returned

The bottom pane shows the SQL query being edited:

```
LAST_NAME,  
EXP  
FROM  
emp_record_table  
WHERE  
EXP > 10;
```

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 GetEmployeesWithExperienceMoreThanThreeYears()

BEGIN

SELECT

EMP_ID,

FIRST_NAME,

LAST_NAME,

EXP

FROM

emp_record_table

WHERE

EXP > 3;

END //

DELIMITER ;

CALL GetEmployeesWithExperienceMoreThanThreeYears();

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
120 WHERE
121     EXP > 3;
122 END //
123 DELIMITER ;
124
125
126 • CALL GetEmployeesWithExperienceMoreThanThreeYears();
127
```

The result grid displays the following data:

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

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
110 DELIMITER //
111 • CREATE PROCEDURE GetEmployeesWithExperienceMoreThanThreeYears()
112 BEGIN
113     SELECT
114         EMP_ID,
115         FIRST_NAME,
116         LAST_NAME,
117         EXP
```

The result grid displays the following data:

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

SQL File 9* SCIENCE_TECH_PROJECT_S... data_science_team emp_record_table proj_ SQLA

```

117      EXP
118      FROM
119      emp_record_table
120      WHERE
121      EXP > 3;
122  END //
123  DELIMITER ;
124

```

Result Grid

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

Result 39 x Read Only

Output

Action Output

#	Time	Action	Message
93	01:55:25	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, RANK() OVER (ORD...	19 row(s) returned
94	01:57:47	CREATE VIEW high_salary_employees AS SELECT EMP_ID, FIRST_NAME, LA...	Error Code: 1050.
95	01:57:56	SELECT * FROM high_salary_employees LIMIT 0, 1000	12 row(s) returned
96	02:00:21	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_t...	8 row(s) returned
97	02:01:44	CREATE PROCEDURE GetEmployeesWithExperienceMoreThanThreeYears() BEGIN S...	Error Code: 1304.
98	02:01:50	CALL GetEmployeesWithExperienceMoreThanThreeYears()	15 row(s) returned

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.

```

SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    EXP,
    GetJobProfile(EXP) AS Job_Profile
FROM
    emp_record_table
WHERE
    DEPT = 'Data Science';

```

```
130 • SELECT
131     EMP_ID,
132     FIRST_NAME,
133     LAST_NAME,
134     EXP,
135     GetJobProfile(EXP) AS Job_Profile
136 FROM
137     emp_record_table
138 WHERE
139     DEPT = 'Data Science';
140
```

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_firstname ON employee (FIRST_NAME);
EXPLAIN SELECT * FROM employee WHERE FIRST_NAME = 'Eric';
```

```
SQL File 9*  SCIENCE_TECH_PROJECT_S_  data_science_team  emp_record_table  proj_
140
141 • CREATE INDEX idx_firstname ON employee (FIRST_NAME);
142 • CREATE INDEX idx_firstname ON your_database.employee (FIRST_NAME);
143
```

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,
    (0.05 * SALARY * EMP_RATING) AS Bonus
FROM
    emp_record_table;
```

SQL File 9* SCIENCE_TECH_PROJECT_S... data_science_team emp_record_table proj_...

Limit to 1000 rows

```

143
144 SELECT
145 EMP_ID,
146 FIRST_NAME,
147 LAST_NAME,
148 SALARY,
149 EMP_RATING,
150 (0.05 * SALARY * EMP_RATING) AS Bonus

```

Result Grid

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

Result 40 x

Output

#	Time	Action	Message
95	01:57:56	SELECT * FROM high_salary_employees LIMIT 0, 1000	12 row(s) returned
96	02:00:21	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_t...	8 row(s) returned
97	02:01:44	CREATE PROCEDURE GetEmployeesWithExperienceMoreThanThreeYears() BEGIN S...	Error Code: 1304. I
98	02:01:50	CALL GetEmployeesWithExperienceMoreThanThreeYears()	15 row(s) returned
99	02:03:32	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, GetJobProfile(EXP) A...	Error Code: 1305. I
100	02:05:08	SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, ...	19 row(s) returned

SQL File 9* SCIENCE_TECH_PROJECT_S... data_science_team emp_record_table proj_...

Limit to 1000 rows

```

147 LAST_NAME,
148 SALARY,
149 EMP_RATING,
150 (0.05 * SALARY * EMP_RATING) AS Bonus
151 FROM
152 emp_record_table;
153
154

```

Result Grid

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

Result 40 x

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;
```

The screenshot shows the SQL Developer interface with a query window titled 'SCIENCE_TECH_PROJECT_S...'. The query is as follows:

```
155 SELECT  
156     CONTINENT,  
157     COUNTRY,  
158     AVG(SALARY) AS Average_Salary  
159 FROM  
160     emp_record_table  
161 GROUP BY  
162     CONTINENT, COUNTRY;
```

Below the query window is the 'Result Grid' showing the following data:

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

Below the result grid is the 'Output' window showing the 'Action Output' table:

#	Time	Action	Message
96	02:00:21	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_t...	8 row(s) returned
97	02:01:44	CREATE PROCEDURE GetEmployeesWithExperienceMoreThanThreeYears() BEGIN S...	Error Code: 1304
98	02:01:50	CALL GetEmployeesWithExperienceMoreThanThreeYears()	15 row(s) returned
99	02:03:32	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, GetJobProfile(EXP) A...	Error Code: 1305
100	02:05:08	SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, ...	19 row(s) returned
101	02:06:16	SELECT CONTINENT, COUNTRY, AVG(SALARY) AS Average_Salary FROM e...	7 row(s) returned

This screenshot is identical to the one above, showing the same SQL query and its result grid. The query is:

```
156 CONTINENT,  
157 COUNTRY,  
158 AVG(SALARY) AS Average_Salary  
159 FROM  
160     emp_record_table  
161 GROUP BY  
162     CONTINENT, COUNTRY;
```

The result grid shows the same data as before:

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

All Action output screenshot: -

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓	4 23:25:28	SELECT * FROM ore.cricket LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec
✓	5 23:25:48	SELECT * FROM ore.product_list LIMIT 0, 1000	5 row(s) returned	0.031 sec / 0.000 sec
✗	6 23:33:49	create table customer(id int primary key, 'name' varchar(50) not null, 'age' int not null, 'city' c...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds...	0.016 sec
✓	7 23:35:17	CREATE TABLE customer (id INT PRIMARY KEY, name VARCHAR(50) NOT NULL, ...	0 row(s) affected	0.047 sec
✓	8 23:37:03	select * from customer LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec
✓	9 23:41:04	Apply changes to CUST1	Changes applied	
✓	10 23:43:09	SELECT * FROM testdb.cust1 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
✓	11 23:49:00	insert into customer (id,name,age,city,salary) VALUES (1,'RAM',24,'PUNE',8000), (2,'SHYA...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.016 sec
✓	12 23:49:22	SELECT * FROM customer LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
✓	13 23:52:20	update customer SET name = 'GOURAV', age = 28 where id = 3	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
✓	14 23:52:50	SELECT * FROM customer LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
✓	15 00:18:24	CREATE DATABASE IF NOT EXISTS employee	1 row(s) affected	0.032 sec
⚠	16 00:19:17	CREATE DATABASE IF NOT EXISTS employee	1 row(s) affected, 1 warning(s): 1007 Can't create database 'employee'; database exists	0.015 sec
✓	17 00:20:02	DROP DATABASE 'employee'	0 row(s) affected	0.047 sec
✓	18 00:20:08	CREATE DATABASE IF NOT EXISTS employee	1 row(s) affected	0.031 sec
✓	19 00:21:45	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓	22 00:21:47	SHOW TABLES FROM 'employee' like 'data_science_team'	OK	0.000 sec
✓	23 00:21:53	CREATE TABLE 'employee`.`data_science_team` ('EMP_ID' text, 'FIRST_NAME' text, 'LA...	OK	0.000 sec
✓	24 00:21:53	PREPARE stmt FROM 'INSERT INTO 'employee`.`data_science_team` ('EMP_ID','FIRST_N...	OK	0.000 sec
✓	25 00:21:53	DEALLOCATE PREPARE stmt	OK	0.000 sec
✓	26 00:22:19	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
✓	27 00:22:19	SHOW DATABASES	OK	0.000 sec
✓	28 00:22:20	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
✓	29 00:22:20	SHOW TABLES FROM 'employee' like 'emp_record_table'	OK	0.000 sec
✓	30 00:22:22	CREATE TABLE 'employee`.`emp_record_table` ('EMP_ID' text, 'FIRST_NAME' text, 'LAS...	OK	0.000 sec
✓	31 00:22:22	PREPARE stmt FROM 'INSERT INTO 'employee`.`emp_record_table` ('EMP_ID','FIRST_N...	OK	0.000 sec
✓	32 00:22:23	DEALLOCATE PREPARE stmt	OK	0.000 sec
✓	33 00:22:46	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
✓	34 00:22:46	SHOW DATABASES	OK	0.000 sec
✓	35 00:22:47	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
✓	36 00:22:47	SHOW TABLES FROM 'employee' like 'proj_table'	OK	0.000 sec
✓	37 00:22:50	CREATE TABLE 'employee`.`proj_table` ('PROJECT_ID' text, 'PROJ_NAME' text, 'DOMAI...	OK	0.000 sec

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓	37 00:22:50	CREATE TABLE 'employee`.`proj_table` ('PROJECT_ID' text, 'PROJ_NAME' text, 'DOMAI...	OK	0.000 sec
✓	38 00:22:50	PREPARE stmt FROM 'INSERT INTO 'employee`.`proj_table` ('PROJECT_ID','PROJ_NAM...	OK	0.000 sec
✓	39 00:22:50	DEALLOCATE PREPARE stmt	OK	0.000 sec
✗	40 00:25:22	DESCRIBE emp_record_table	Error Code: 1146. Table 'testdb.emp_record_table' doesn't exist	0.000 sec
✗	41 00:25:43	DESCRIBE emp_record_table	Error Code: 1146. Table 'testdb.emp_record_table' doesn't exist	0.000 sec
✗	42 00:28:55	create table data_science_team	Error Code: 4028. A table must have at least one visible column.	0.000 sec
✓	43 00:29:08	SELECT * FROM employee.data_science_team LIMIT 0, 1000	13 row(s) returned	0.000 sec / 0.000 sec
✓	44 00:29:11	SELECT * FROM employee.emp_record_table LIMIT 0, 1000	19 row(s) returned	0.000 sec / 0.000 sec
✓	45 00:29:15	SELECT * FROM employee.proj_table LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
✗	46 00:30:38	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_tabl...	Error Code: 1146. Table 'testdb.emp_record_table' doesn't exist	0.000 sec
✓	47 00:31:38	SHOW TABLES	2 row(s) returned	0.016 sec / 0.000 sec
✓	48 00:32:22	DESCRIBE emp_record_table	13 row(s) returned	0.000 sec / 0.000 sec
✓	49 00:32:40	DESCRIBE Proj_table	7 row(s) returned	0.016 sec / 0.000 sec
✓	50 00:32:55	DESCRIBE Data_science_team	9 row(s) returned	0.000 sec / 0.000 sec
✓	51 00:33:03	DESCRIBE Proj_table	7 row(s) returned	0.000 sec / 0.000 sec
✓	52 00:33:47	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT FROM emp_record_tabl...	19 row(s) returned	0.000 sec / 0.000 sec

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓ 86	01:47:20	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATIN...	4 row(s) returned	0.000 sec / 0.000 sec
✓ 87	01:47:58	SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, EMP_RATIN...	12 row(s) returned	0.000 sec / 0.000 sec
✓ 88	01:48:34	SELECT CONCAT(FIRST_NAME, '', LAST_NAME) AS NAME FROM emp_record_table W...	0 row(s) returned	0.000 sec / 0.000 sec
✓ 89	01:49:25	SELECT E1.EMP_ID AS Manager_ID, CONCAT(E1.FIRST_NAME, '', E1.LAST_NA...	6 row(s) returned	0.000 sec / 0.000 sec
✓ 90	01:50:39	SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPARTMENT FROM emp_record_table ...	0 row(s) returned	0.000 sec / 0.000 sec
✓ 91	01:52:43	SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, ...	19 row(s) returned	0.000 sec / 0.000 sec
✓ 92	01:53:51	SELECT ROLE, MIN(SALARY) AS Min_Salary, MAX(SALARY) AS Max_Salary FR...	6 row(s) returned	0.015 sec / 0.000 sec
✓ 93	01:55:25	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, RANK() OVER (ORD...	19 row(s) returned	0.000 sec / 0.000 sec
✗ 94	01:57:47	CREATE VIEW high_salary_employees AS SELECT EMP_ID, FIRST_NAME, LA...	Error Code: 1050. Table 'high_salary_employees' already exists	0.000 sec
✓ 95	01:57:56	SELECT * FROM high_salary_employees LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
✓ 96	02:00:21	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP FROM emp_record_t...	8 row(s) returned	0.000 sec / 0.000 sec
✗ 97	02:01:44	CREATE PROCEDURE GetEmployeesWithExperienceMoreThanThreeYears() BEGIN S...	Error Code: 1304. PROCEDURE GetEmployeesWithExperienceMoreThanThreeYears alrea...	0.000 sec
✓ 98	02:01:50	CALL GetEmployeesWithExperienceMoreThanThreeYears()	15 row(s) returned	0.000 sec / 0.000 sec
✗ 99	02:03:32	SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP, GetJobProfile(EXP) A...	Error Code: 1305. FUNCTION employee.GetJobProfile does not exist	0.000 sec
✓ 100	02:05:08	SELECT EMP_ID, FIRST_NAME, LAST_NAME, SALARY, EMP_RATING, ...	19 row(s) returned	0.000 sec / 0.000 sec
✓ 101	02:06:16	SELECT CONTINENT, COUNTRY, AVG(SALARY) AS Average_Salary FROM e...	7 row(s) returned	0.016 sec / 0.000 sec

SQL Training

100% of Self-Learning Completed | Projects completed: 0/3

Notes

Help

MySQL-workbench

This Lab will get reset on 22nd April 2024, 5:58 AM

Key	Value	Action
VM DNS Name	<p>Looks like either the virtual machine or RDP gateway isn't running.</p> <p>Looks like either the virtual machine or RDP gateway isn't running. We have initiated the power-on operation, this typically takes upto 5 minutes. Please try launching the VM access again in 5 minutes. You can also monitor the progress using "Resources" tab.</p>	
VM Admin Username		
VM Admin Password		

OK