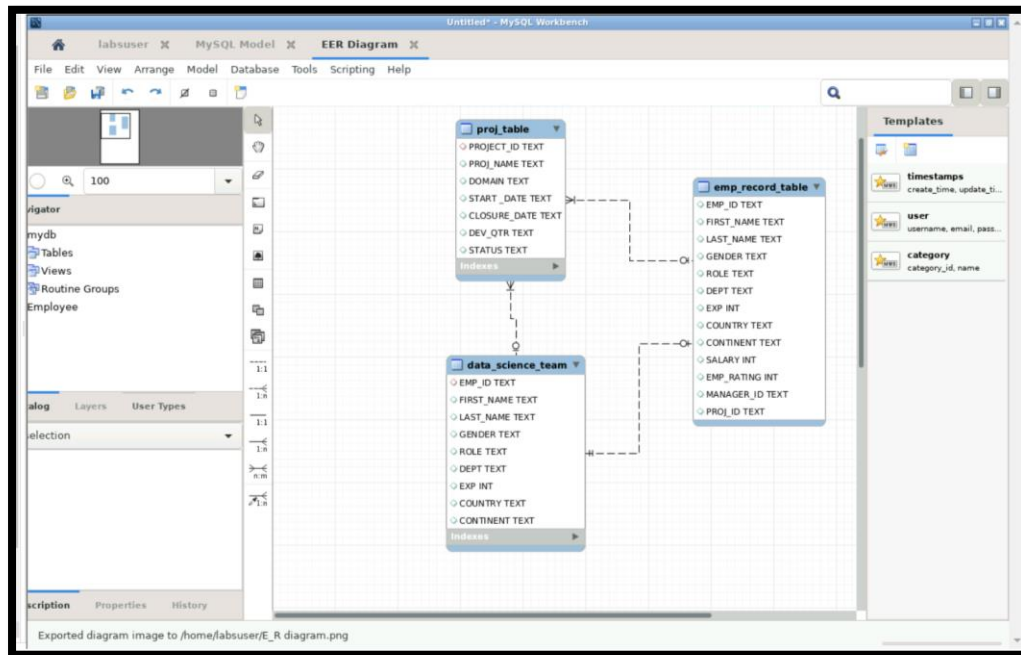


ScienceQtech Employee Performance Mapping

MySQL – Project Screen Shot

E-R DIAGRAM:



Screen shot - Query

```
1 • SELECT * FROM Employee.emp_record_table;
2 • use Employee;
3 • SELECT EMP_ID,FIRST_NAME, LAST_NAME,DEPT,EMP_RATING from emp_record_table where EMP_RATING < 2;
4 • SELECT EMP_ID,FIRST_NAME, LAST_NAME,DEPT,EMP_RATING from emp_record_table where EMP_RATING > 4;
5 • SELECT EMP_ID,FIRST_NAME, LAST_NAME,DEPT,EMP_RATING from emp_record_table where EMP_RATING between 2 and 4;
6 • select concat(first_name,last_name) as Name,dept from emp_record_table where dept = 'FINANCE';
7 • select emp_id,first_name,last_name,role from emp_record_table where emp_id in (select distinct (E.manager_id)
8 • select emp_id,first_name,last_name,dept from emp_record_table where dept = 'Finance' union
9 • select E.emp_id,E.first_name,E.last_name,E.dept from emp_record_table E where dept = 'Healthcare');
10 • select emp_id,first_name,last_name,dept,emp_rating,max(emp_rating) over(partition by dept)as Dept_max_rating
11 • from emp_record_table;
12 • select role,min(salary)as Min_salary,max(salary) as Max_salary
13 • from emp_record_table group by role;
14 • Delimiter $$
15 • create procedure get_empl_exp()
16 • begin
17 • select emp_id,concat(first_name,last_name)as name,dept,exp from emp_record_table where exp>3;
18 • end $$
19 • call get_empl_exp();
20 •
21 • delimiter $$
22 • create function role_check(exp integer)
23 • returns varchar(225) deterministic
24 • BEGIN declare role varchar(225);
25 • if exp<=2 then set role='junior data scientist';
26 • elseif exp between 2 and 5 then set role='Associate data scientist';
27 • elseif exp between 5 and 10 then set role = 'senior data scientist';
```

QUERY OUTPUT:

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

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

The screenshot shows the SQL Developer interface with a query window titled 'employee_project'. The query is: `1 * SELECT * FROM Employee.emp_record_table;`
`2 * use Employee;`
`3 * SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, EMP_RATING from emp_record_table where EMP_RATING < 2;`

The result grid displays the following data:

#	EMP_ID	FIRST_NAME	LAST_NAME	DEPT	EMP_RATING
1	E057	Dorothy	Wilson	HEALTHCARE	1
2	E532	Claire	Brennan	AUTOMOTIVE	1
3	E620	Katrina	Allen	RETAIL	1

The screenshot shows the SQL Developer interface with a query window titled 'employee_project'. The query is: `3 * SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, EMP_RATING from emp_record_table where EMP_RATING < 2;`
`4 * SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, EMP_RATING from emp_record_table where EMP_RATING > 4;`
`5 * SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, EMP_RATING from emp_record_table where EMP_RATING between 2 and 4;`

The result grid displays the following data:

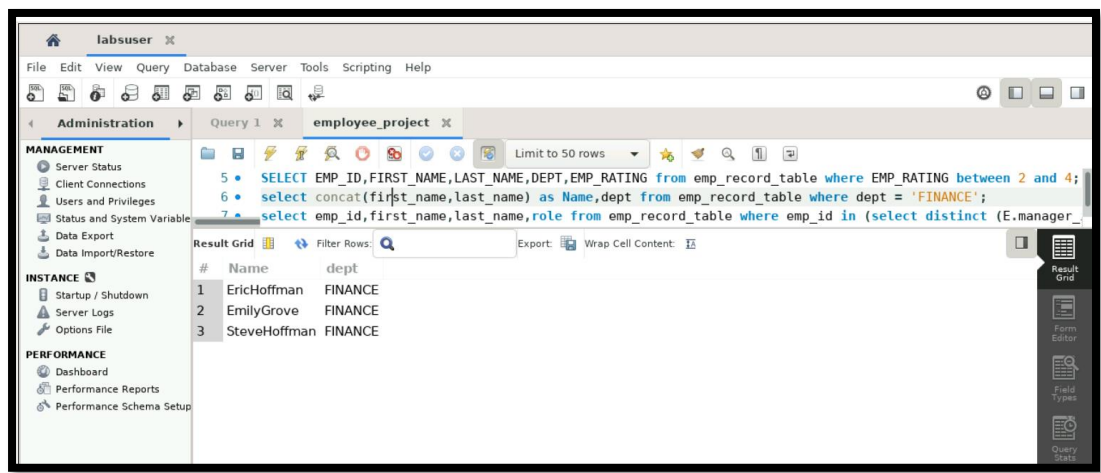
#	EMP_ID	FIRST_NAME	LAST_NAME	DEPT	EMP_RATING
1	E001	Arthur	Black	ALL	5
2	E052	Dianna	Wilson	HEALTHCARE	5
3	E083	Patrick	Voltz	HEALTHCARE	5
4	E204	Karene	Nowak	AUTOMOTIVE	5

The screenshot shows the SQL Developer interface with a query window titled 'employee_project'. The query is: `4 * SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, EMP_RATING from emp_record_table where EMP_RATING > 4;`
`5 * SELECT EMP_ID, FIRST_NAME, LAST_NAME, DEPT, EMP_RATING from emp_record_table where EMP_RATING between 2 and 4;`
`6 * select concat(first_name, last_name) as Name, dept from emp_record_table where dept = 'FINANCE';`

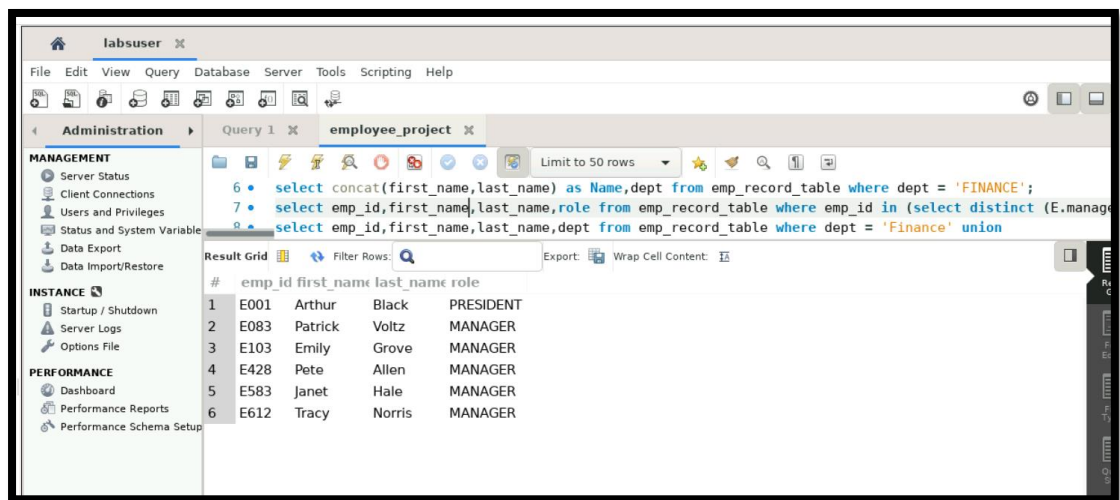
The result grid displays the following data:

#	EMP_ID	FIRST_NAME	LAST_NAME	DEPT	EMP_RATING
1	E005	Eric	Hoffman	FINANCE	3
2	E010	William	Butler	AUTOMOTIVE	2
3	E103	Emily	Grove	FINANCE	4
4	E245	Nian	Zhen	RETAIL	2
5	E260	Roy	Collins	RETAIL	3
6	E403	Steve	Hoffman	FINANCE	3
7	E428	Pete	Allen	AUTOMOTIVE	4
8	E478	David	Smith	RETAIL	4
9	E505	Chad	Wilson	HEALTHCARE	2
10	E583	Janet	Hale	RETAIL	2
11	E612	Tracy	Norris	RETAIL	4
12	E640	Jennifer	Jones	RETAIL	4

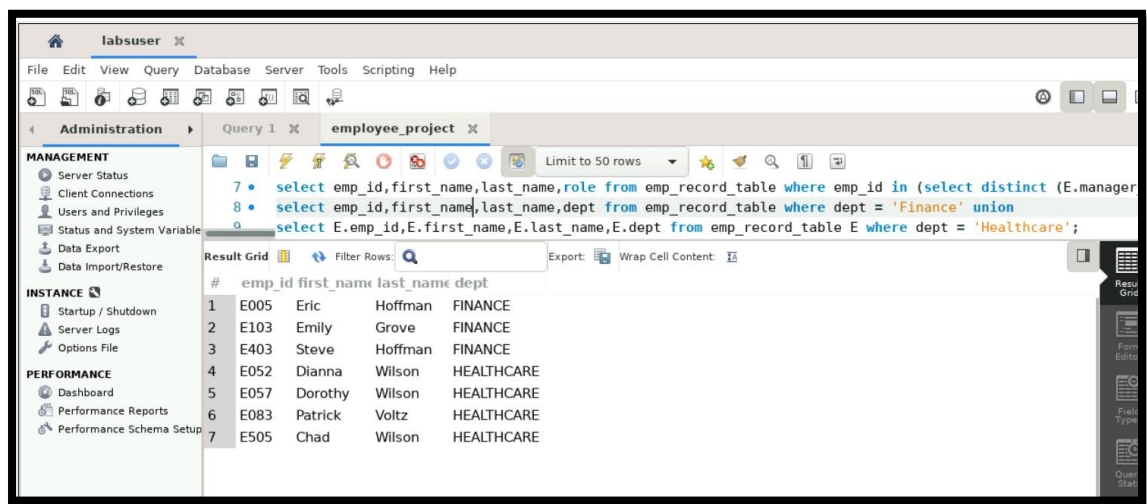
Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME.



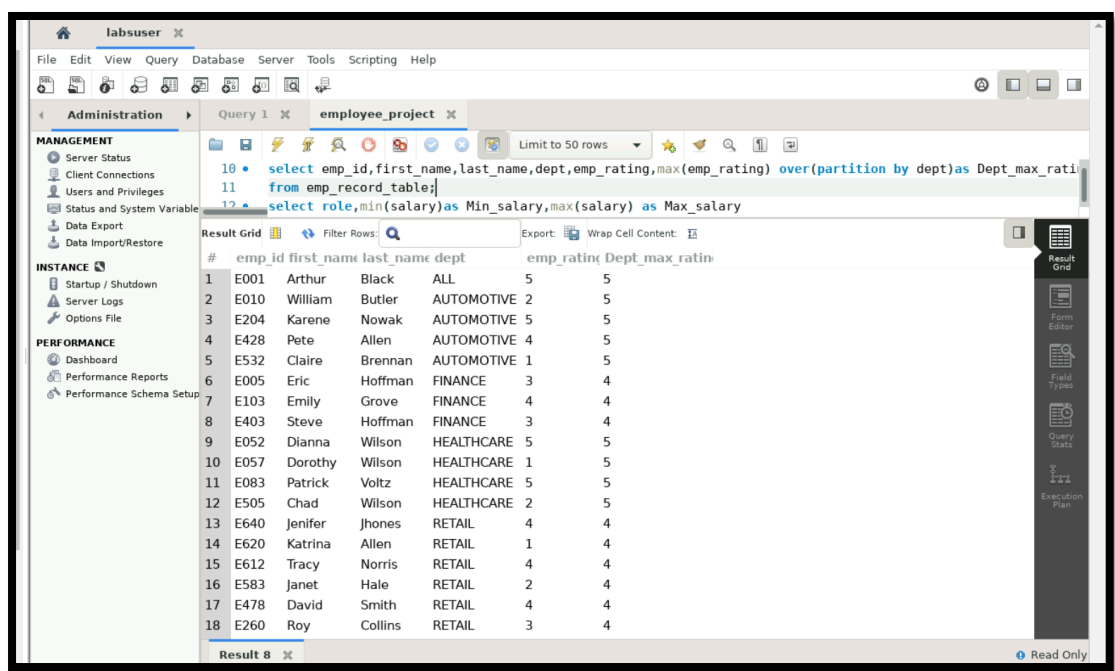
Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).



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



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



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

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

```

12 • select role,min(salary)as Min_salary,max(salary) as Max_salary
13 • from emp_record_table group by role;
14 • Delimiter $$

```

The result grid displays the following data:

#	role	Min_salary	Max_salary
1	PRESIDENT	16500	16500
2	LEAD DATA SCIENTIST	8500	9000
3	SENIOR DATA SCIENTIST	5500	7700
4	MANAGER	8500	11000
5	ASSOCIATE DATA SCIENTIST	4000	5000
6	JUNIOR DATA SCIENTIST	2800	3000

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

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

```

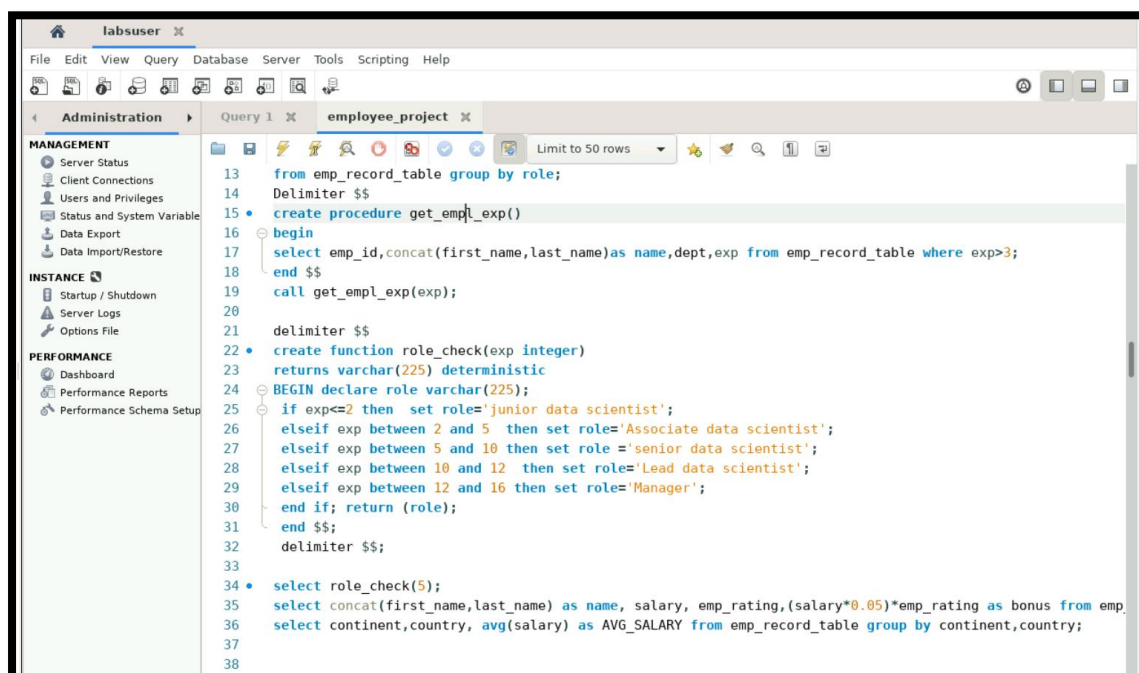
10 • select emp_id,concat(first_name,last_name)as NAME,role,dept,exp,rank() over(order by exp DESC)as exp_rank
11 • from emp_record_table;
12 • select emp_id,first_name,last_name,dept,emp_rating,max(emp_rating) over(partition by dept)as Dept_max_rating

```

The result grid displays the following data:

#	emp_id	NAME	role	dept	exp	exp_ranking
1	E001	ArthurBlack	PRESIDENT	ALL	20	1
2	E083	PatrickVoltz	MANAGER	HEALTHCARE	15	2
3	E103	EmilyGrove	MANAGER	FINANCE	14	3
4	E583	JanetHale	MANAGER	RETAIL	14	3
5	E428	PeteAllen	MANAGER	AUTOMOTIVE	14	3
6	E612	TracyNorris	MANAGER	RETAIL	13	6
7	E010	WilliamButler	LEAD DATA SCIENTIST	AUTOMOTIVE	12	7
8	E005	EricHoffman	LEAD DATA SCIENTIST	FINANCE	11	8
9	E057	DorothyWilson	SENIOR DATA SCIENTIST	HEALTHCARE	9	9
10	E204	KareneNowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	10
11	E260	RoyCollins	SENIOR DATA SCIENTIST	RETAIL	7	11
12	E245	NianZhen	SENIOR DATA SCIENTIST	RETAIL	6	12
13	E052	DiannaWilson	SENIOR DATA SCIENTIST	HEALTHCARE	6	12
14	E505	ChadWilson	ASSOCIATE DATA SCIE...	HEALTHCARE	5	14
15	E403	SteveHoffman	ASSOCIATE DATA SCIE...	FINANCE	4	15
16	E478	DavidSmith	ASSOCIATE DATA SCIE...	RETAIL	3	16
17	E532	ClaireBrennan	ASSOCIATE DATA SCIE...	AUTOMOTIVE	3	16
18	E620	KatrinaAllen	JUNIOR DATA SCIENTIST	RETAIL	2	18

1. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table
2. 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 is:
 - 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'
3. 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).
4. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table



```
13 from emp_record_table group by role;
14 Delimiter $$
15 • create procedure get_empl_exp()
16 begin
17 select emp_id,concat(first_name,last_name)as name,dept,exp from emp_record_table where exp>3;
18 end $$
19 call get_empl_exp(exp);
20
21 delimiter $$
22 • create function role_check(exp integer)
23 returns varchar(225) deterministic
24 BEGIN declare role varchar(225);
25 if exp<=2 then set role='junior data scientist';
26 elseif exp between 2 and 5 then set role='Associate data scientist';
27 elseif exp between 5 and 10 then set role='senior data scientist';
28 elseif exp between 10 and 12 then set role='Lead data scientist';
29 elseif exp between 12 and 16 then set role='Manager';
30 end if; return (role);
31 end $$;
32 delimiter $$;
33
34 • select role_check(5);
35 select concat(first_name,last_name) as name, salary, emp_rating,(salary*0.05)*emp_rating as bonus from emp
36 select continent,country, avg(salary) as AVG_SALARY from emp_record_table group by continent,country;
37
38
```


labsuser

File Edit View Query Database Server Tools Scripting Help

Administration Query 1 employee_project* Limit to 50 rows

```
34 • select role_check(5);
35
36 select concat(first_name,last_name) as name, salary, emp_rating,(salary*0.05)*emp_rating as bonus from emp
```

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

#	continent	country	AVG_SALARY
1	NORTH AMERICA	USA	9440.0000
2	EUROPE	FRANCE	9000.0000
3	NORTH AMERICA	CANADA	7000.0000
4	EUROPE	GERMANY	7600.0000
5	ASIA	CHINA	6500.0000
6	ASIA	INDIA	6166.6667
7	SOUTH AMERICA	COLOMBIA	5600.0000

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variable
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Result Grid Form Editor Field Types Query Stats Execution Plan

labsuser

File Edit View Query Database Server Tools Scripting Help

Administration Query 1 employee_project* Limit to 50 rows

```
34 • select role_check(5);
35
36 select concat(first_name,last_name) as name, salary, emp_rating,(salary*0.05)*emp_rating as bonus from emp
```

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

#	name	salary	emp_rating	bonus
1	ArthurBlack	16500	5	4125.00
2	EricHoffman	8500	3	1275.00
3	WilliamButler	9000	2	900.00
4	DiannaWilson	5500	5	1375.00
5	DorothyWilson	7700	1	385.00
6	PatrickVoltz	9500	5	2375.00
7	EmilyGrove	10500	4	2100.00
8	KareneNowak	7500	5	1875.00
9	NianZhen	6500	2	650.00
10	RoyCollins	7000	3	1050.00
11	SteveHoffman	5000	3	750.00
12	PeteAllen	11000	4	2200.00
13	DavidSmith	4000	4	800.00
14	ChadWilson	5000	2	500.00
15	ClaireBrennan	4300	1	215.00
16	JanetHale	10000	2	1000.00
17	TracyNorris	8500	4	1700.00
18	KatrinaAllen	3000	1	150.00

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variable
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Result Grid Form Editor Field Types Query Stats Execution Plan