

# ScienceQtech Employee Performance Mapping

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.

SQL code: CREATE DATABASE employee;

SQL code: CREATE TABLE emp\_record (  
emp\_id VARCHAR(6) not null PRIMARY KEY,  
f\_name VARCHAR(10) not null,  
l\_name VARCHAR(10) not null,  
gender VARCHAR(10) not null,  
role VARCHAR(30) not null,  
dept VARCHAR(15) not null,  
exp INT not null,  
country VARCHAR(15) not null,  
continent VARCHAR(15) not null,  
salary INT not null,  
emp\_rating INT not null,  
manager\_id VARCHAR(5),  
proj\_id varchar(5));

DESCRIBE emp\_record;

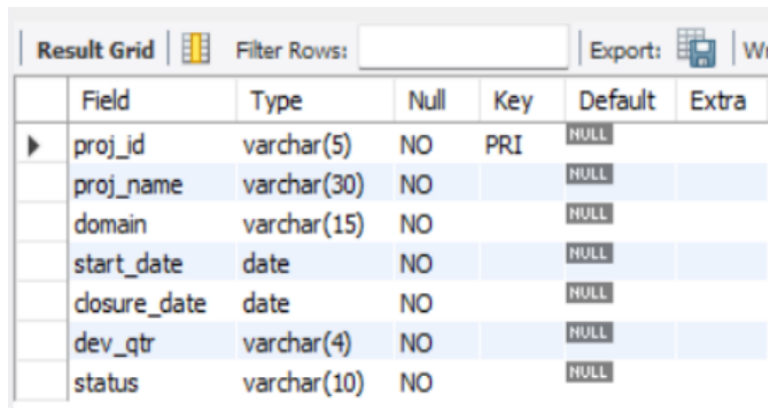
Result Grid		Filter Rows:		Export:		
	Field	Type	Null	Key	Default	Extra
►	emp_id	varchar(6)	NO	PRI	NULL	
	f_name	varchar(10)	NO		NULL	
	l_name	varchar(10)	NO		NULL	
	gender	varchar(10)	NO		NULL	
	role	varchar(30)	NO		NULL	
	dept	varchar(15)	NO		NULL	
	exp	int	NO		NULL	
	country	varchar(15)	NO		NULL	
	continent	varchar(15)	NO		NULL	
	salary	int	NO		NULL	
	emp_rating	int	NO		NULL	
	manager...	varchar(5)	YES		NULL	
	proj_id	varchar(5)	YES		NULL	

SQL code:

```
CREATE TABLE proj_table (  
proj_id VARCHAR(5) not null PRIMARY KEY,  
proj_name VARCHAR(30) not null,  
domain VARCHAR(15) not null,  
start_date DATE not null,  
closure_date DATE not null,  
dev_qtr VARCHAR(4) not null,  
status VARCHAR(10) not null);
```

DESCRIBE proj\_table;

Output:



The screenshot shows a database tool interface with a 'Result Grid' tab selected. The grid displays the structure of the 'proj\_table' as described in the SQL code. The columns are Field, Type, Null, Key, Default, and Extra. The rows correspond to the table's attributes: proj\_id (varchar(5), NO, PRI, NULL), proj\_name (varchar(30), NO, NULL), domain (varchar(15), NO, NULL), start\_date (date, NO, NULL), closure\_date (date, NO, NULL), dev\_qtr (varchar(4), NO, NULL), and status (varchar(10), NO, NULL).

	Field	Type	Null	Key	Default	Extra
▶	proj_id	varchar(5)	NO	PRI	NULL	
	proj_name	varchar(30)	NO		NULL	
	domain	varchar(15)	NO		NULL	
	start_date	date	NO		NULL	
	closure_date	date	NO		NULL	
	dev_qtr	varchar(4)	NO		NULL	
	status	varchar(10)	NO		NULL	

SQL code:

```
CREATE TABLE data_sci_team (  
emp_id VARCHAR(6) not null PRIMARY KEY,  
f_name VARCHAR(10) not null,  
l_name VARCHAR(10) not null,  
gender VARCHAR(10) not null,  
role VARCHAR(30) not null,  
dept VARCHAR(15) not null,  
exp INT not null,  
country VARCHAR(15) not null,  
continent VARCHAR(15) not null);
```

DESCRIBE data\_sci\_team;

Output :

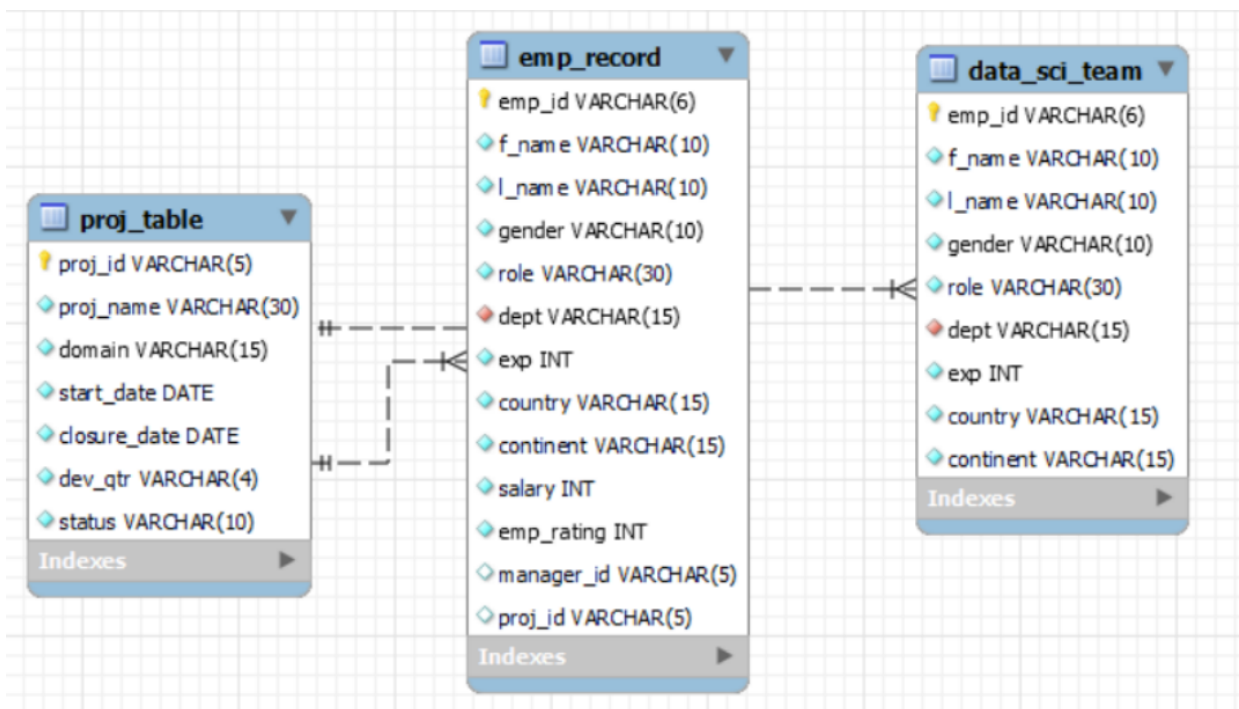


## data\_science\_team

emp_id	f_name	l_name	gender	role	dept	exp	country	continent
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA
E057	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA
E204	Karene	Nowak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE
E245	Nian	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA
E260	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA
E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA
E532	Claire	Brennan	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	GERMANY	EUROPE
E620	Katrina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA
E640	Jenifer	Jhones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Create an ER diagram for the given employee database.  
Using reverse engineering.

Output:





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.

SQL code:

```
SELECT emp_id, f_name, l_name, gender, dept  
FROM emp_record;
```

Output:

Result Grid				Filter Rows:	Edit:
	emp_id	f_name	l_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
	E505	Chad	Wilson	M	HEALTHCARE
	E532	Claire	Brennan	F	AUTOMOTIVE
	E583	Janet	Hale	F	RETAIL
	E612	Tracy	Norris	F	RETAIL
	E620	Katrina	Allen	F	RETAIL
	E640	Jenifer	Jhones	F	RETAIL
✱	NULL	NULL	NULL	NULL	NULL

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

EMP\_RATING is Less than two

SQL code:

```
SELECT emp_id, f_name, l_name, gender, dept, emp_rating  
FROM emp_record  
WHERE emp_rating < 2;
```

Output:

Result Grid		Filter Rows:		Edit:		
	emp_id	f_name	l_name	gender	dept	emp_rating
▶	E057	Dorothy	Wilson	F	HEALTHCARE	1
	E532	Claire	Brennan	F	AUTOMOTIVE	1
	E620	Katrina	Allen	F	RETAIL	1
✱	NULL	NULL	NULL	NULL	NULL	NULL

EMP\_RATING is Greater than four

SQL code:

```
SELECT emp_id, f_name, l_name, gender, dept, emp_rating
FROM emp_record
WHERE emp_rating > 4;
```

Output:

Result Grid		Filter Rows:		Edit:		
	emp_id	f_name	l_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
✱	NULL	NULL	NULL	NULL	NULL	NULL

EMP\_RATING is Between two and four

SQL code:

```
SELECT emp_id, f_name, l_name, gender, dept, emp_rating
FROM emp_record
WHERE emp_rating BETWEEN 2 AND 4;
```

Output:

	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.

SQL code:

```
SELECT CONCAT(f_name,' ',l_name) NAME
FROM emp_record
WHERE dept = 'FINANCE';
```

Output :

Result Grid	
	NAME
▶	Eric Hoffman
	Emily Grove
	Steve Hoffman

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

SQL code:

```
SELECT role, manager_id, count(*)
FROM emp_record
GROUP BY manager_id
ORDER BY manager_id;
```

Result Grid			
	role	manager_id	count(*)
▶	PRESIDENT	NULL	1
	MANAGER	E001	5
	SENIOR DATA SCIENTIST	E083	3
	LEAD DATA SCIENTIST	E103	2
	LEAD DATA SCIENTIST	E428	3
	SENIOR DATA SCIENTIST	E583	3
	JUNIOR DATA SCIENTIST	E612	2







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

SQL code:

```
SELECT role, MIN(salary) AS minSalary, MAX(salary) AS maxSalary
FROM emp_record
GROUP BY role;
```

Result Grid			
Filter Rows:			
Export:			
Wrap			
	role	minSalary	maxSalary
▶	PRESIDENT	16500	16500
	LEAD DATA SCIENTIST	8500	9000
	SENIOR DATA SCIENTIST	5500	7700
	MANAGER	8500	11000
	ASSOCIATE DATA SCIENTIST	4000	5000
	JUNIOR DATA SCIENTIST	2800	3000

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

SQL code:

```
SELECT f_name, l_name, exp as experience,
DENSE_RANK() OVER (ORDER BY exp DESC) exp_rank
FROM emp_record;
```

Result Grid				
Filter Rows:				
Export:				
Wrap				
	f_name	l_name	experience	exp_rank
▶	Arthur	Black	20	1
	Patrick	Voltz	15	2
	Emily	Grove	14	3
	Pete	Allen	14	3
	Janet	Hale	14	3
	Tracy	Norris	13	4
	William	Butler	12	5
	Eric	Hoffman	11	6
	Dorothy	Wilson	9	7
	Karene	Nowak	8	8
	Roy	Collins	7	9
	Dianna	Wilson	6	10
	Nian	Zhen	6	10
	Chad	Wilson	5	11
	Steve	Hoffman	4	12
	David	Smith	3	13
	Claire	Brennan	3	13
	Katrina	Allen	2	14
	Jenifer	Jhones	1	15

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.

SQL code:

```
CREATE VIEW 6K_salary AS
```

```
SELECT emp_id, f_name, l_name, country, salary
```

```
FROM emp_record
```

```
WHERE salary > 6000;
```

```
SELECT * FROM 6k_salary;
```

Result Grid					
Filter Rows:					
	emp_id	f_name	l_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

SQL code:

```
SELECT emp_id, f_name, l_name, exp
```

```
FROM emp_record
```

```
WHERE exp IN ( SELECT exp FROM emp_record WHERE exp > 10 );
```

Result Grid				
Filter Rows:				
	emp_id	f_name	l_name	exp
▶	E001	Arthur	Black	20
	E005	Eric	Hoffman	11
	E010	William	Butler	12
	E083	Patrick	Voltz	15
	E103	Emily	Grove	14
	E428	Pete	Allen	14
	E583	Janet	Hale	14
	E612	Tracy	Norris	13
*	NULL	NULL	NULL	NULL

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

```
SQL code:
DELIMITER //
CREATE PROCEDURE Employee3()
BEGIN
    SELECT * FROM emp_record
    WHERE exp > 3;
END //
DELIMITER ;
CALL Employee3;
```

emp_id	f_name	l_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
E052	Dianna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083	P103
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
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E103	P105
E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4	E001	NULL
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
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

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'.

```
SQL code:
DELIMITER //
CREATE PROCEDURE check_role()
BEGIN
    SELECT * FROM emp_record
    CASE
        WHEN exp <= 2 THEN SET role = 'JUNIOR DATA SCIENTIST';
        WHEN exp BETWEEN 3 AND 5 THEN SET role = 'ASSOCIATE DATA SCIENTIST';
        WHEN exp BETWEEN 6 AND 10 THEN SET role = 'SENIOR DATA SCIENTIST';
        WHEN exp BETWEEN 11 AND 12 THEN SET role = 'LEAD DATA SCIENTIST';
```

```

        WHEN exp BETWEEN 13 AND 16 THEN SET role = 'MANAGER';
        ELSE SET role = 'all good';
    END CASE;
END //
DELIMITER ;

```

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.

SQL code:

```
ALTER TABLE emp_record ADD INDEX fname_index (f_name);
```

```
SELECT * FROM emp_record WHERE f_name = 'Eric';
```

Output												
Action Output												
#	Time	Action	Message							Duration / Fetch		
1	08:51:11	alter table emp_record add index fname_index (f_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0							0.047 sec		
emp_id	f_name	l_name	gender	role	dept	exp	country	continent	salary	emp_rating	manager_id	proj_id
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E103	P105

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).

SQL code:

```
SELECT f_name, l_name, salary, ((salary * .05)*emp_rating) AS bonus
```

```
FROM emp_record;
```

Result Grid				
Filter Rows:				
	f_name	l_name	salary	bonus
▶	Arthur	Black	16500	4125.00
	Eric	Hoffman	8500	1275.00
	William	Butler	9000	900.00
	Dianna	Wilson	5500	1375.00
	Dorothy	Wilson	7700	385.00
	Patrick	Voltz	9500	2375.00
	Emily	Grove	10500	2100.00
	Karene	Nowak	7500	1875.00
	Nian	Zhen	6500	650.00
	Roy	Collins	7000	1050.00
	Steve	Hoffman	5000	750.00
	Pete	Allen	11000	2200.00
	David	Smith	4000	800.00
	Chad	Wilson	5000	500.00
	Claire	Brennan	4300	215.00
	Janet	Hale	10000	1000.00
	Tracy	Norris	8500	1700.00
	Katrina	Allen	3000	150.00
	Jenifer	Jhones	2800	560.00

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

Average salary based on the continent

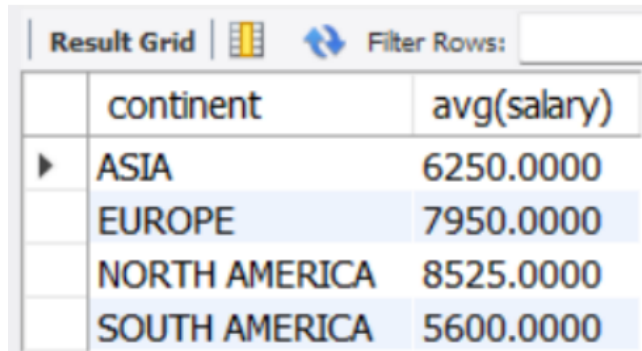
SQL code:

```
SELECT continent, AVG(salary)
```

```
FROM emp_record
```

```
GROUP BY continent
```

```
ORDER BY continent ASC;
```



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'continent' and 'avg(salary)'. The data is sorted by continent in ascending order. The rows are: ASIA (6250.0000), EUROPE (7950.0000), NORTH AMERICA (8525.0000), and SOUTH AMERICA (5600.0000). A 'Filter Rows:' input field is visible at the top right of the grid.

	continent	avg(salary)
▶	ASIA	6250.0000
	EUROPE	7950.0000
	NORTH AMERICA	8525.0000
	SOUTH AMERICA	5600.0000

Average salary based on the country

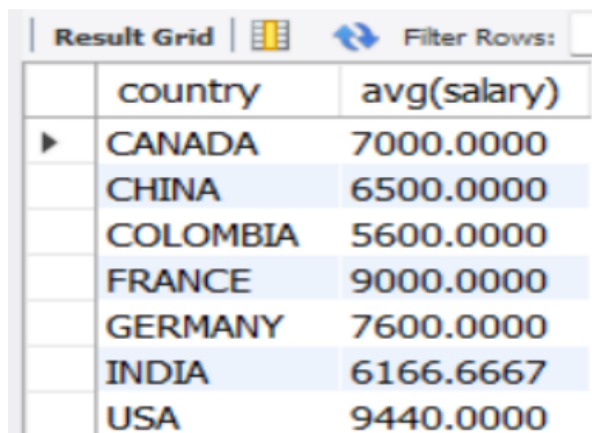
SQL code:

```
SELECT country, AVG(salary)
```

```
FROM emp_record
```

```
GROUP BY country
```

```
ORDER BY country ASC;
```



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'country' and 'avg(salary)'. The data is sorted by country in ascending order. The rows are: CANADA (7000.0000), CHINA (6500.0000), COLOMBIA (5600.0000), FRANCE (9000.0000), GERMANY (7600.0000), INDIA (6166.6667), and USA (9440.0000). A 'Filter Rows:' input field is visible at the top right of the grid.

	country	avg(salary)
▶	CANADA	7000.0000
	CHINA	6500.0000
	COLOMBIA	5600.0000
	FRANCE	9000.0000
	GERMANY	7600.0000
	INDIA	6166.6667
	USA	9440.0000