

ScienceQtech writeup

Course-end Project 2 of PGC DS Foundations: SQL

The task to be performed:

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 database *employee*;

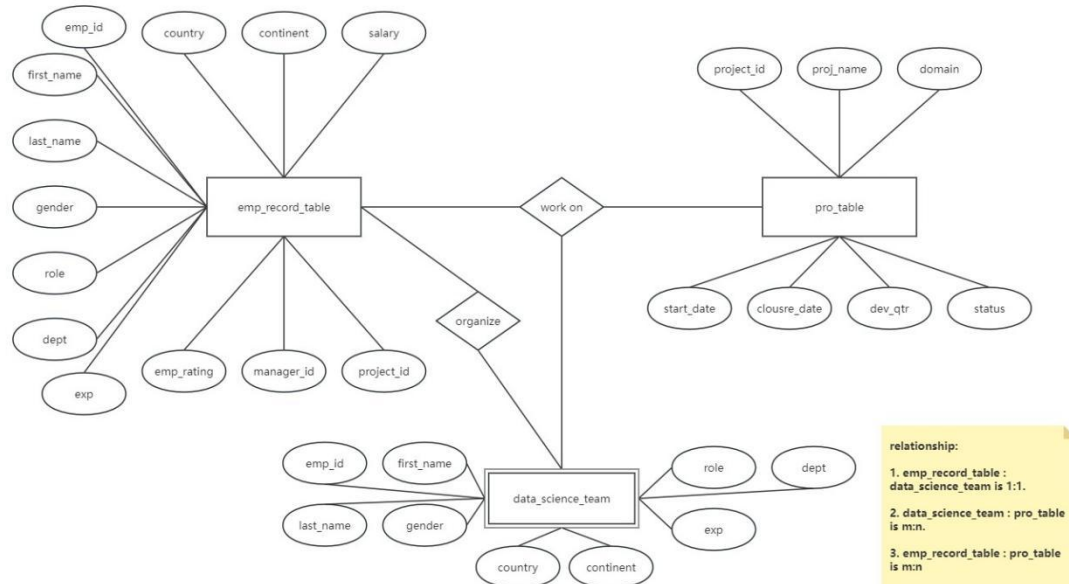
Use table data import wizard to import the three CSV files.

Furthermore, alter the three tables by using the following code:

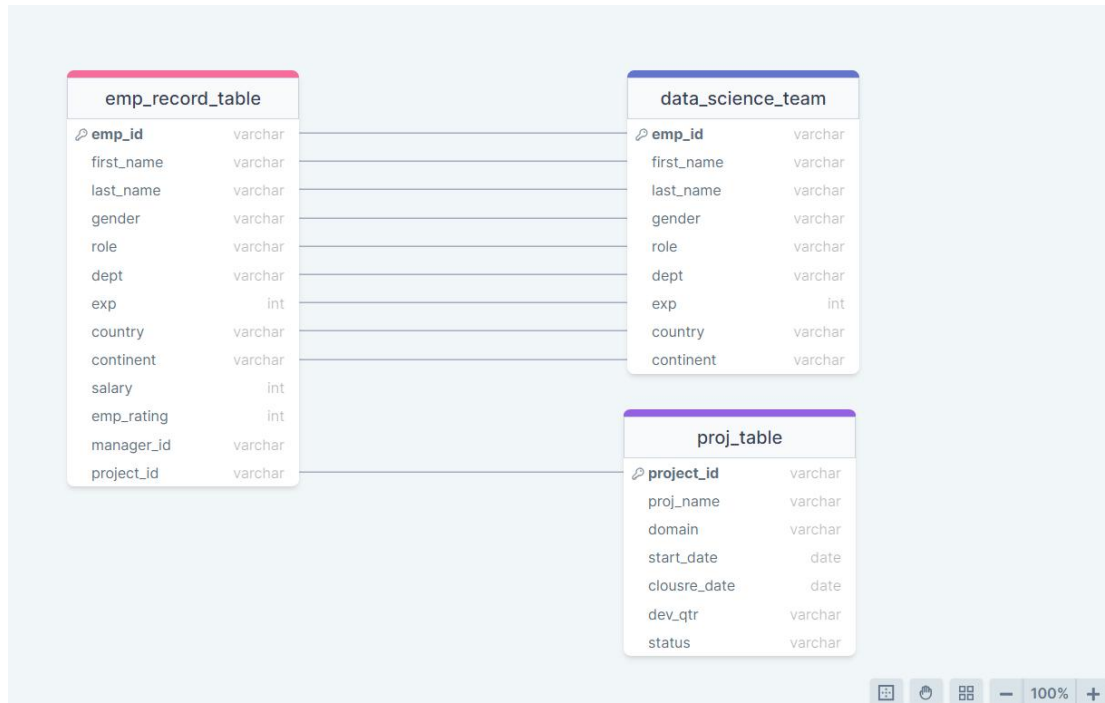
```
ALTER TABLE `employee`.`data_science_team`  
CHANGE COLUMN `EMP_ID` `EMP_ID` VARCHAR(100) NOT NULL ,  
CHANGE COLUMN `FIRST_NAME` `FIRST_NAME` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `LAST_NAME` `LAST_NAME` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `GENDER` `GENDER` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `ROLE` `ROLE` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `DEPT` `DEPT` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `COUNTRY` `COUNTRY` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `CONTINENT` `CONTINENT` VARCHAR(100) NULL DEFAULT NULL ,  
ADD PRIMARY KEY (`EMP_ID`);  
;  
ALTER TABLE `employee`.`emp_record_table`  
CHANGE COLUMN `EMP_ID` `EMP_ID` VARCHAR(100) NOT NULL ,  
CHANGE COLUMN `FIRST_NAME` `FIRST_NAME` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `LAST_NAME` `LAST_NAME` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `GENDER` `GENDER` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `ROLE` `ROLE` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `DEPT` `DEPT` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `COUNTRY` `COUNTRY` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `CONTINENT` `CONTINENT` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `MANAGER_ID` `MANAGER_ID` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `PROJ_ID` `PROJ_ID` VARCHAR(100) NULL DEFAULT NULL ,  
ADD PRIMARY KEY (`EMP_ID`);  
;  
ALTER TABLE `employee`.`proj_table`  
CHANGE COLUMN `PROJECT_ID` `PROJECT_ID` VARCHAR(100) NOT NULL ,  
CHANGE COLUMN `PROJ_NAME` `PROJ_NAME` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `DOMAIN` `DOMAIN` VARCHAR(100) NULL DEFAULT NULL ,  
CHANGE COLUMN `DEV_QTR` `DEV_QTR` VARCHAR(100) NULL DEFAULT NULL ,
```

CHANGE COLUMN `STATUS` `STATUS` VARCHAR(100) NULL DEFAULT NULL ,
 ADD PRIMARY KEY (`PROJECT_ID`);
 ;

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



pic 1 ER diagram



pic 2 UML diagram

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;
```

	emp_id	first_name	last_name	gender	dept
▶	E001	Arthur	Black	M	ALL
	E005	Eric	Hoffman	M	FINANCE
	E010	William	Butler	M	AUTOMOTIVE
	E052	Dianna	Wilson	F	HEALTHCARE
	E057	Dorothy	Wilson	F	HEALTHCARE
	E083	Patrick	Voltz	M	HEALTHCARE
	E103	Emily	Grove	F	FINANCE
	E204	Karene	Nowak	F	AUTOMOTIVE
	E245	Nian	Zhen	M	RETAIL
	E260	Roy	Collins	M	RETAIL
	E403	Steve	Hoffman	M	FINANCE
	E428	Pete	Allen	M	AUTOMOTIVE
	E478	David	Smith	M	RETAIL
	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

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

```
select emp_id, first_name, last_name, gender, dept, emp_rating from emp_record_table
where emp_rating<2;
```

	emp_id	first_name	last_name	gender	dept	emp_rating
▶	E057	Dorothy	Wilson	F	HEALTHCARE	1
	E532	Claire	Brennan	F	AUTOMOTIVE	1
	E620	Katrina	Allen	F	RETAIL	1

```
select emp_id, first_name, last_name, gender, dept, emp_rating from emp_record_table
where emp_rating>4;
```

	emp_id	first_name	last_name	gender	dept	emp_rating
►	E001	Arthur	Black	M	ALL	5
	E052	Dianna	Wilson	F	HEALTHCARE	5
	E083	Patrick	Voltz	M	HEALTHCARE	5
	E204	Karene	Nowak	F	AUTOMOTIVE	5

SELECT emp_id, first_name, last_name, gender, dept, emp_rating FROM emp_record_table
WHERE emp_rating between 2 and 4;

	emp_id	first_name	last_name	gender	dept	emp_rating
►	E005	Eric	Hoffman	M	FINANCE	3
	E010	William	Butler	M	AUTOMOTIVE	2
	E103	Emily	Grove	F	FINANCE	4
	E245	Nian	Zhen	M	RETAIL	2
	E260	Roy	Collins	M	RETAIL	3
	E403	Steve	Hoffman	M	FINANCE	3
	E428	Pete	Allen	M	AUTOMOTIVE	4
	E478	David	Smith	M	RETAIL	4
	E505	Chad	Wilson	M	HEALTHCARE	2
	E583	Janet	Hale	F	RETAIL	2
	E612	Tracy	Norris	F	RETAIL	4
	E640	Jenifer	Jhones	F	RETAIL	4

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

SELECT CONCAT(emp_record_table.first_name,' ', emp_record_table.last_name) AS name
FROM emp_record_table WHERE dept="finance";

	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).

SELECT m.emp_id, m.first_name, m.last_name, m.role, m.dept, COUNT(e.emp_id) AS
no_direct_report
FROM emp_record_table m
INNER JOIN emp_record_table e
ON m.emp_id = e.manager_id
GROUP BY m.emp_id
ORDER BY m.emp_id;

	emp_id	first_name	last_name	role	dept	no_direct_report
▶	E001	Arthur	Black	PRESIDENT	ALL	5
	E083	Patrick	Voltz	MANAGER	HEALTHCARE	3
	E103	Emily	Grove	MANAGER	FINANCE	2
	E428	Pete	Allen	MANAGER	AUTOMOTIVE	3
	E583	Janet	Hale	MANAGER	RETAIL	3
	E612	Tracy	Norris	MANAGER	RETAIL	2

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

```
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';
```

	emp_id	first_name	dept
▶	E052	Dianna	HEALTHCARE
	E057	Dorothy	HEALTHCARE
	E083	Patrick	HEALTHCARE
	E505	Chad	HEALTHCARE
	E005	Eric	FINANCE
	E103	Emily	FINANCE
	E403	Steve	FINANCE

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, dept, emp_rating, MAX(emp_rating) OVER
(PARTITION BY dept) AS Max_Dept_Rating
FROM emp_record_table
ORDER BY dept, emp_rating DESC;
```

	emp_id	first_name	last_name	role	dept	emp_rating	Max_Dept_Rating
▶	E001	Arthur	Black	PRESIDENT	ALL	5	5
	E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
	E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
	E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
	E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
	E103	Emily	Grove	MANAGER	FINANCE	4	4
	E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4
	E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
	E052	Dianna	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	5	5
	E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	5
	E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	HEALTHCARE	2	5
	E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	HEALTHCARE	1	5
	E478	David	Smith	ASSOCIATE DATA SCIENTIST	RETAIL	4	4
	E612	Tracy	Norris	MANAGER	RETAIL	4	4
	E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	RETAIL	4	4
	E260	Roy	Collins	SENIOR DATA SCIENTIST	RETAIL	3	4
	E245	Nian	Zhen	SENIOR DATA SCIENTIST	RETAIL	2	4
	E583	Janet	Hale	MANAGER	RETAIL	2	4
	E620	Katrina	Allen	JUNIOR DATA SCIENTIST	RETAIL	1	4

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

```
SELECT role, MIN(salary) AS minimun_salary, MAX(salary) AS max_salary FROM emp_record_table GROUP BY role;
```

	role	minimun_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

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, role, exp, dense_rank() over (order by exp) as exp_rank_of_employee from emp_record_table;
```

	emp_id	first_name	last_name	role	exp	exp_rank_of_employee
►	E640	Jenifer	Jhones	JUNIOR DATA SCIENTIST	1	1
	E620	Katrina	Allen	JUNIOR DATA SCIENTIST	2	2
	E478	David	Smith	ASSOCIATE DATA SCIENTIST	3	3
	E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	3	3
	E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	4	4
	E505	Chad	Wilson	ASSOCIATE DATA SCIENTIST	5	5
	E052	Dianna	Wilson	SENIOR DATA SCIENTIST	6	6
	E245	Nian	Zhen	SENIOR DATA SCIENTIST	6	6
	E260	Roy	Collins	SENIOR DATA SCIENTIST	7	7
	E204	Karene	Nowak	SENIOR DATA SCIENTIST	8	8
	E057	Dorothy	Wilson	SENIOR DATA SCIENTIST	9	9
	E005	Eric	Hoffman	LEAD DATA SCIENTIST	11	10
	E010	William	Butler	LEAD DATA SCIENTIST	12	11
	E612	Tracy	Norris	MANAGER	13	12
	E103	Emily	Grove	MANAGER	14	13
	E428	Pete	Allen	MANAGER	14	13
	E583	Janet	Hale	MANAGER	14	13
	E083	Patrick	Voltz	MANAGER	15	14
	E001	Arthur	Black	PRESIDENT	20	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.

```
create view employee_country_view as
select emp_id, country, salary from emp_record_table where salary > 6000;
SELECT * FROM employee.employee_country_view;
```

	emp_id	country	salary
▶	E001	USA	16500
	E005	USA	8500
	E010	FRANCE	9000
	E057	USA	7700
	E083	USA	9500
	E103	CANADA	10500
	E204	GERMANY	7500
	E245	CHINA	6500
	E260	INDIA	7000
	E428	GERMANY	11000
	E583	COLOMBIA	10000
	E612	INDIA	8500

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, dept, exp from emp_record_table
where exists(Select * from emp_record_table where exp > 10) and exp > 10;
```

	emp_id	first_name	last_name	dept	exp
▶	E001	Arthur	Black	ALL	20
	E005	Eric	Hoffman	FINANCE	11
	E010	William	Butler	AUTOMOTIVE	12
	E083	Patrick	Voltz	HEALTHCARE	15
	E103	Emily	Grove	FINANCE	14
	E428	Pete	Allen	AUTOMOTIVE	14
	E583	Janet	Hale	RETAIL	14
	E612	Tracy	Norris	RETAIL	13

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

```
delimiter $$
create procedure get_experience()
begin
select * from emp_record_table where exp > 3;
end $$
call get_experience();
```

	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
	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'.

DELIMITER \$\$

CREATE FUNCTION employee.get_job_profile(exp INT)

RETURNS VARCHAR(2000) DETERMINISTIC

BEGIN

DECLARE job_profile VARCHAR(2000);

IF exp <= 2 THEN

SET job_profile = 'JUNIOR DATA SCIENTIST';

ELSEIF exp <= 5 THEN

SET job_profile = 'ASSOCIATE DATA SCIENTIST';

ELSEIF exp <= 10 THEN

SET job_profile = 'SENIOR DATA SCIENTIST';

ELSEIF exp <= 12 THEN

SET job_profile = 'LEAD DATA SCIENTIST';

ELSEIF exp <= 16 THEN

SET job_profile = 'MANAGER';

END IF;

RETURN job_profile;

END \$\$

SELECT first_name, last_name, exp, role, get_job_profile(exp) AS Employee_profile

FROM emp_record_table order by exp;

	first_name	last_name	exp	role	Employee_profile
▶	Jenifer	Jhones	1	JUNIOR DATA SCIENTIST	JUNIOR DATA SCIENTIST
	Katrina	Allen	2	JUNIOR DATA SCIENTIST	JUNIOR DATA SCIENTIST
	David	Smith	3	ASSOCIATE DATA SCIENTIST	ASSOCIATE DATA SCIENTIST
	Claire	Brennan	3	ASSOCIATE DATA SCIENTIST	ASSOCIATE DATA SCIENTIST
	Steve	Hoffman	4	ASSOCIATE DATA SCIENTIST	ASSOCIATE DATA SCIENTIST
	Chad	Wilson	5	ASSOCIATE DATA SCIENTIST	ASSOCIATE DATA SCIENTIST
	Dianna	Wilson	6	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST
	Nian	Zhen	6	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST
	Roy	Collins	7	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST
	Karene	Nowak	8	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST
	Dorothy	Wilson	9	SENIOR DATA SCIENTIST	SENIOR DATA SCIENTIST
	Eric	Hoffman	11	LEAD DATA SCIENTIST	LEAD DATA SCIENTIST
	William	Butler	12	LEAD DATA SCIENTIST	LEAD DATA SCIENTIST
	Tracy	Norris	13	MANAGER	MANAGER
	Emily	Grove	14	MANAGER	MANAGER
	Pete	Allen	14	MANAGER	MANAGER
	Janet	Hale	14	MANAGER	MANAGER
	Patrick	Voltz	15	MANAGER	MANAGER
	Arthur	Black	20	PRESIDENT	NULL

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_first_name on emp_record_table(First_name);
select * from emp_record_table where First_name = 'Eric';
```

	EMP_ID	FIRST_NAME	LAST_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
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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 order by BONUS;
```

	emp_id	first_name	last_name	salary	emp_rating	BONUS
▶	E620	Katrina	Allen	3000	1	150.00
	E532	Claire	Brennan	4300	1	215.00
	E057	Dorothy	Wilson	7700	1	385.00
	E505	Chad	Wilson	5000	2	500.00
	E640	Jenifer	Jhones	2800	4	560.00
	E245	Nian	Zhen	6500	2	650.00
	E403	Steve	Hoffman	5000	3	750.00
	E478	David	Smith	4000	4	800.00
	E010	William	Butler	9000	2	900.00
	E583	Janet	Hale	10000	2	1000.00
	E260	Roy	Collins	7000	3	1050.00
	E005	Eric	Hoffman	8500	3	1275.00
	E052	Dianna	Wilson	5500	5	1375.00
	E612	Tracy	Norris	8500	4	1700.00
	E204	Karene	Nowak	7500	5	1875.00
	E103	Emily	Grove	10500	4	2100.00
	E428	Pete	Allen	11000	4	2200.00
	E083	Patrick	Voltz	9500	5	2375.00
	E001	Arthur	Black	16500	5	4125.00

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

```
select emp_id, first_name, last_name, country, continent,
avg(salary) over ( partition by country) AVG_Salary_Distr_Country,
avg(salary) over ( partition by continent) AVG_Salary_Distr_Continent from
emp_record_table;
```

	emp_id	first_name	last_name	country	continent	AVG_Salary_Distr_Country	AVG_Salary_Distr_Continent
▶	E245	Nian	Zhen	CHINA	ASIA	6500.0000	6250.0000
	E260	Roy	Collins	INDIA	ASIA	6166.6667	6250.0000
	E612	Tracy	Norris	INDIA	ASIA	6166.6667	6250.0000
	E620	Katrina	Allen	INDIA	ASIA	6166.6667	6250.0000
	E010	William	Butler	FRANCE	EUROPE	9000.0000	7950.0000
	E204	Karene	Nowak	GERMANY	EUROPE	7600.0000	7950.0000
	E428	Pete	Allen	GERMANY	EUROPE	7600.0000	7950.0000
	E532	Claire	Brennan	GERMANY	EUROPE	7600.0000	7950.0000
	E052	Dianna	Wilson	CANADA	NORTH AMERICA	7000.0000	8525.0000
	E103	Emily	Grove	CANADA	NORTH AMERICA	7000.0000	8525.0000
	E505	Chad	Wilson	CANADA	NORTH AMERICA	7000.0000	8525.0000
	E001	Arthur	Black	USA	NORTH AMERICA	9440.0000	8525.0000
	E005	Eric	Hoffman	USA	NORTH AMERICA	9440.0000	8525.0000
	E057	Dorothy	Wilson	USA	NORTH AMERICA	9440.0000	8525.0000
	E083	Patrick	Voltz	USA	NORTH AMERICA	9440.0000	8525.0000
	E403	Steve	Hoffman	USA	NORTH AMERICA	9440.0000	8525.0000
	E478	David	Smith	COLOMBIA	SOUTH AMERICA	5600.0000	5600.0000
	E583	Janet	Hale	COLOMBIA	SOUTH AMERICA	5600.0000	5600.0000
	E640	Jenifer	Jhones	COLOMBIA	SOUTH AMERICA	5600.0000	5600.0000