WRITE UP

1. Create a database named *project* and *employee*, then import **data\_science\_team.csv**and **proj\_table.csv** into the **project** database and **emp\_record\_table.csv** into the **employee** database from the given resources.

create database ScienceQtech\_Database;

CREATE TABLE Proj\_table(PROJ\_ID VARCHAR(5) PRIMARY KEY, PROJ\_NAME VARCHAR(25), DOMAIN VARCHAR(25), START\_DATE DATE, CLOSURE\_DATE DATE, DEV\_QTR VARCHAR(5), SATUS VARCHAR(10) );

create table emp\_record\_table(EMP\_ID varchar(5) PRIMARY KEY, FIRST\_NAME VARCHAR(25), LAST\_NAME VARCHAR(25), GENDER CHAR(1), ROLE VARCHAR(25), DEPT VARCHAR(25), EXP INT, COUNTRY VARCHAR(25), CONTINENT VARCHAR(25), SALARY FLOAT, EMP\_RATING INT, MANAGER\_ID VARCHAR(5), PROJ\_ID VARCHAR(5), FOREIGN KEY(PROJ\_ID) REFERENCES Proj\_table(PROJ\_ID) );

CREATE TABLE Data\_science\_team(EMP\_ID VARCHAR(5), FIRST\_NAME VARCHAR(25), LAST\_NAME VARCHAR(25), GENDER CHAR(1), ROLE VARCHAR(25), DEPT VARCHAR(25), EXP INT, COUNTRY VARCHAR(25), CONTINENT VARCHAR(25) );

1. Create an ER diagram for the given **project** and the **employee** databases.
2. 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;

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

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM emp\_record\_table WHERE EMP\_RATING <2;

* greater than four

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM emp\_record\_table WHERE EMP\_RATING >4;

* between two and four

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM emp\_record\_table WHERE EMP\_RATING >=2 AND EMP\_RATING<=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(FIRST\_NAME,' ', LAST\_NAME) AS NAME FROM emp\_record\_table WHERE DEPT ='FINANCE';

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

select m.first\_name, m.last\_name, m.role, count(r.emp\_id) as Number\_of\_reporters from emp\_record\_table m, emp\_record\_table r where m.emp\_id=r.manager\_id group by m.emp\_id;

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

SELECT \* FROM emp\_record\_table where DEPT='HEALTHCARE' OR DEPT='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\_EMP\_RATING FROM emp\_record\_table;

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, MAX(SALARY) AS MAX\_SALARY, MIN(SALARY) AS MIN\_SALARY from emp\_record\_table group by ROLE;

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

select first\_name, last\_name, role, exp, (case when exp>=15 then 'Executive-level' when exp<15 and exp>=11 then 'Senior' when exp<11 and exp>=7 then 'Mid-level' when exp<7 and exp>=3 then 'Intermediate' else 'Entry-level' end) as ranks from emp\_record\_table;

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 country\_view as select first\_name, last\_name, country, salary from emp\_record\_table where salary>6000;

select \* from country\_view;

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

Select \* from emp\_record\_table where emp\_id in (select emp\_id 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 emp\_exp3()

-> begin

-> select \* from emp\_record\_table where exp>=3;

-> end&&

call emp\_exp3;

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'.

create function check\_job\_profile(exp int, role varchar(25) ) returns varchar(50) deterministic begin declare status varchar(20); if (exp<=2 and role='JUNIOR DATA SCIENTIST') then set status='Checked Standard'; elseif (exp>2 and exp<=5 and role='ASSOCIATE DATA SCIENTIST') then set status='Checked Standard'; elseif (exp>5 and exp<=10 and role='SENIOR DATA SCIENTIST') then set status='Checked Standard'; elseif (exp>10 and exp<=12 and role='LEAD DATA SCIENTIST') then set status='Checked Standard'; elseif (exp>12 and exp<=16 and role='MANAGER') then set status='Checked Standard'; else set status='DO NOT Check Standard'; end if ; return (status); end&&

select first\_name,last\_name,role,exp, check\_job\_profile(exp,role) from data\_science\_team;

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 ind\_first\_name ON emp\_record\_table (first\_name);

select \* from emp\_record\_table where first\_name='Eric';

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 first\_name, last\_name, salary, (salary\*0.05 \* emp\_rating) as Bonus from emp\_record\_table;

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, avg(salary) as Average\_salary\_by\_continent from emp\_record\_table group by continent;

select country, avg(salary) as Average\_salary\_by\_country from emp\_record\_table group by country;