Project on SQL

SCIENCEQTECH

Presentation by Akshay M M

HELLO EVERYONE!

- I have created a project on SQL for ScienceQtech company.
- ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer.

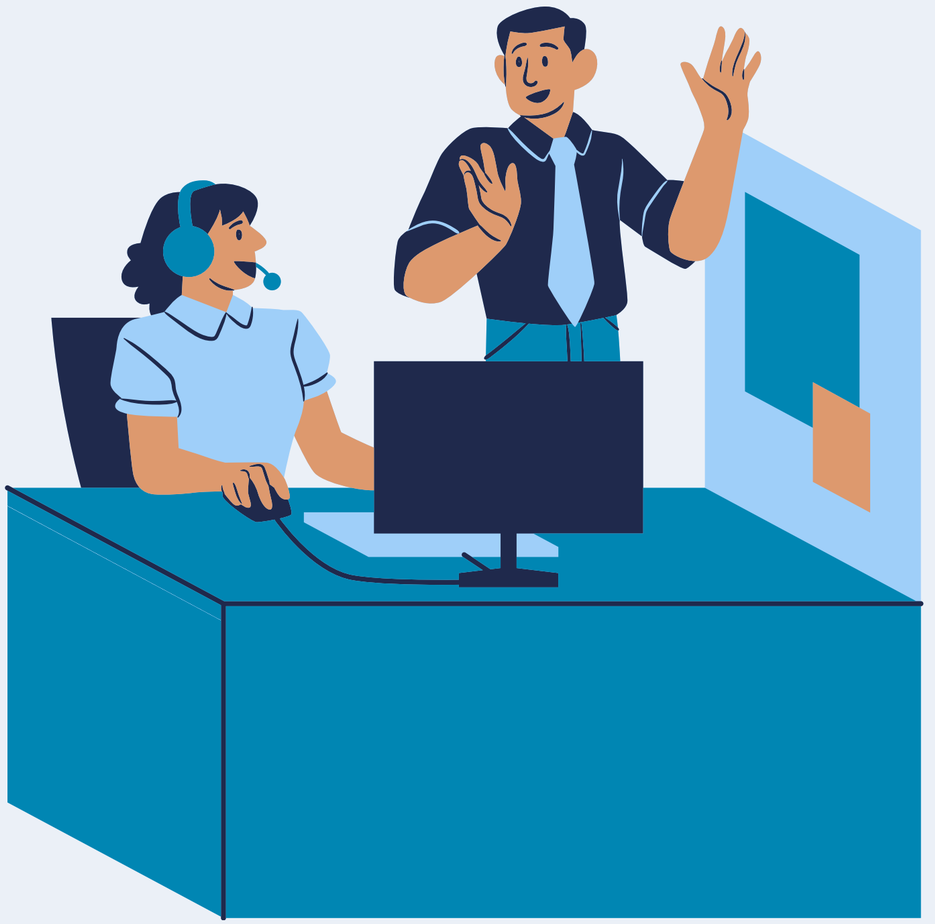
OBJECTIVE

With the annual appraisal cycle around the corner, the HR department has asked to generate reports on employee details, their performance, and on the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements.

DATASET DESCRIPTION

Emp_record_table: It contains the information of all the employees.

- EMP_ID – ID of the employee
- FIRST_NAME – First name of the employee
- LAST_NAME – Last name of the employee
- GENDER – Gender of the employee
- ROLE – Post of the employee
- DEPT – Field of the employee
- EXP – Years of experience the employee has
- COUNTRY – Country in which the employee is presently living
- CONTINENT – Continent in which the country is
- SALARY – Salary of the employee
- EMP_RATING – Performance rating of the employee
- MANAGER_ID – The manager under which the employee is assigned
- PROJ_ID – The project on which the employee is



DATASET DESCRIPTION

Proj_table: It contains information about the projects.

- PROJECT_ID – ID for the project
- PROJ_Name – Name of the project
- DOMAIN – Field of the project
- START_DATE – Day the project began
- CLOSURE_DATE – Day the project was or will be completed
- DEV_QTR – Quarter in which the project was scheduled
- STATUS – Status of the project currently

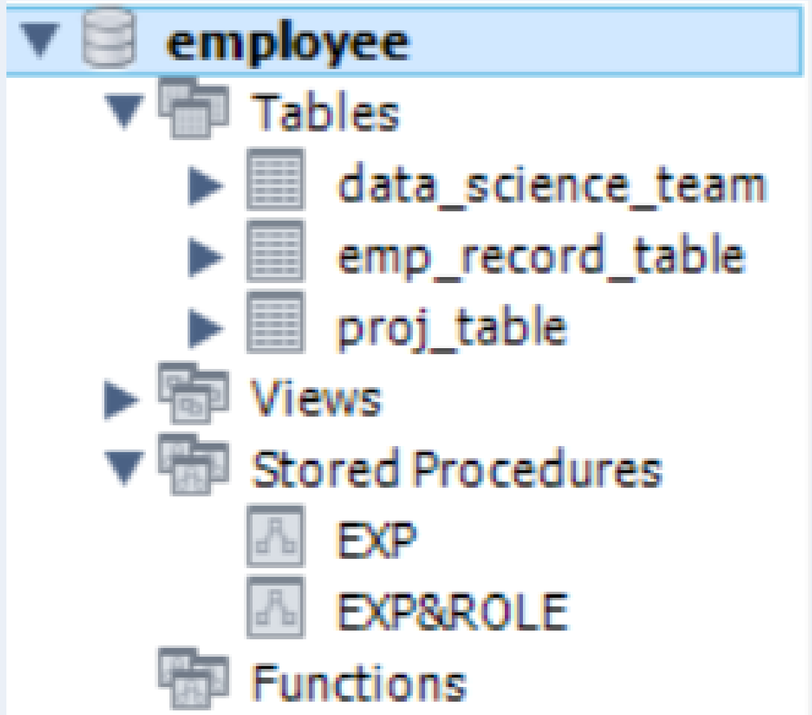
DATASET DESCRIPTION

Data_science_team: It contains information about all the employees in the Data Science team.

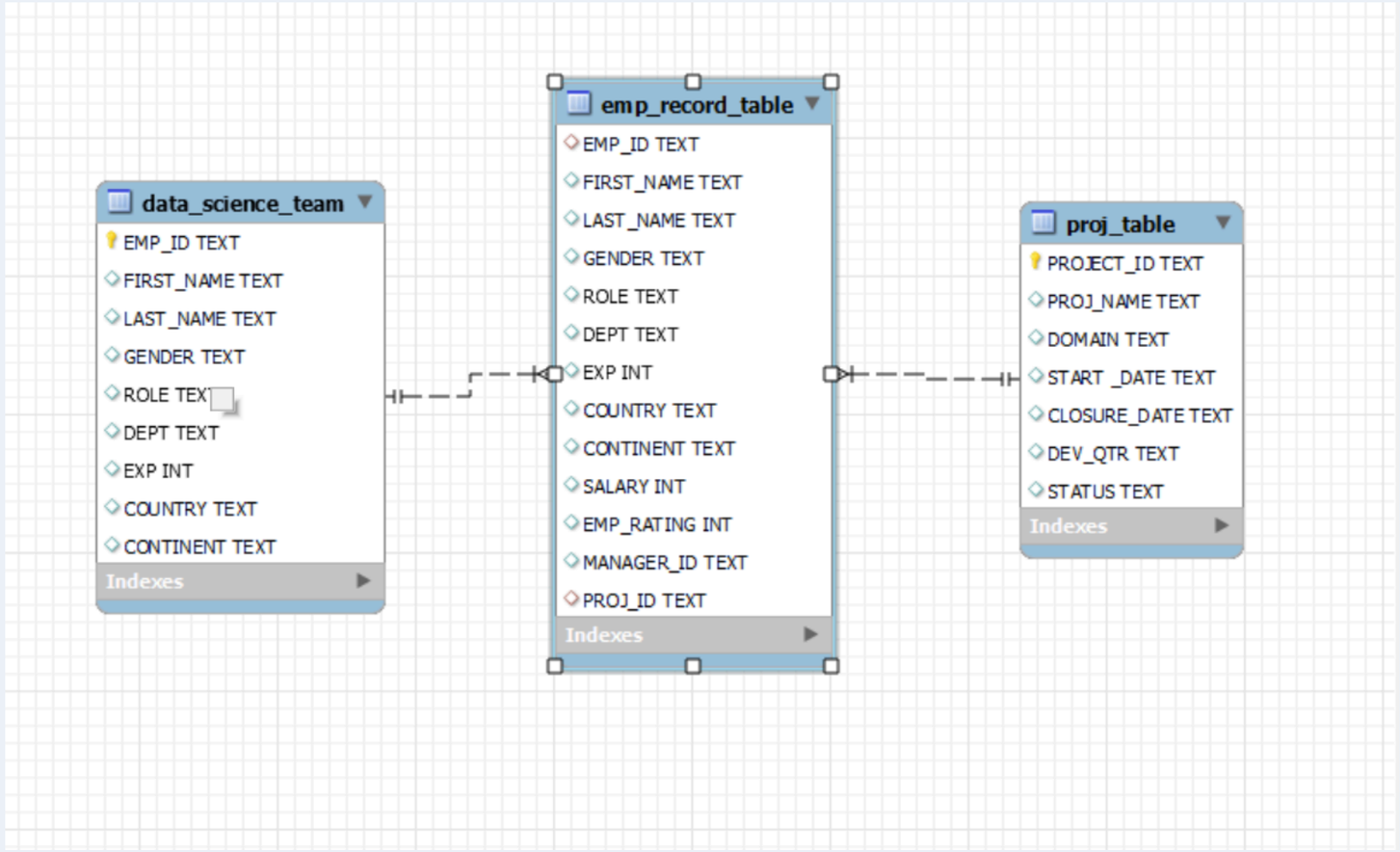
- EMP_ID – ID of the employee
- FIRST_NAME – First name of the employee
- LAST_NAME – Last name of the employee
- GENDER – Gender of the employee
- ROLE – Post of the employee
- DEPT – Field of the employee
- EXP – Years of experience the employee has
- COUNTRY – Country in which the employee is presently living
- CONTINENT – Continent in which the country is

TASKS THAT HAS BEEN PERFORMED

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 AN ER DIAGRAM FOR THE GIVEN EMPLOYEE DATABASE.



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;
```

WRITE A QUERY TO FETCH EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, AND EMP_RATING IF THE EMP_RATING IS:

```sql
SELECT EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT, EMP_RATING
FROM EMP_RECORD_TABLE
WHERE EMP_RATING < 2 OR EMP_RATING > 4 OR (EMP_RATING BETWEEN 2 AND 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.

```sql
SELECT FIRST_NAME, LAST_NAME, CONCAT(FIRST_NAME,' ',LAST_NAME) AS NAME
FROM EMP_RECORD_TABLE
WHERE DEPT = 'FINANCE';
```

WRITE A QUERY TO LIST ONLY THOSE EMPLOYEES WHO HAVE SOMEONE REPORTING TO THEM. ALSO, SHOW THE NUMBER OF REPORTERS (INCLUDING THE PRESIDENT).

```sql
SET SQL_SAFE_UPDATES = 0;
SET SESSION sql_mode = (SELECT REPLACE(@@session.sql_mode, 'ONLY_FULL_GROUP_BY', ''));
SELECT m.EMP_ID, m.FIRST_NAME, m.LAST_NAME, m.ROLE,
m.EXP, COUNT(e.EMP_ID) AS "EMP_COUNT"
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;
```

WRITE A QUERY TO LIST DOWN ALL THE EMPLOYEES FROM THE HEALTHCARE AND FINANCE DEPARTMENTS USING UNION. TAKE DATA FROM THE EMPLOYEE RECORD TABLE.

```sql
SELECT *
FROM EMP_RECORD_TABLE
WHERE DEPT = "FINANCE"
UNION
SELECT *
FROM EMP_RECORD_TABLE
WHERE DEPT = "HEALTHCARE"
ORDER BY DEPT;
```

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)
FROM EMP_RECORD_TABLE
ORDER BY DEPT;
```

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),MIN(SALARY)
FROM EMP_RECORD_TABLE
GROUP BY ROLE;
```

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,
DENSE_RANK() OVER(ORDER BY EXP DESC) AS RANKING
FROM EMP_RECORD_TABLE;
```

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 V_NAME AS
SELECT *
FROM EMP_RECORD_TABLE
WHERE SALARY > 6000;
```

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
                ORDER BY EMP_ID);
```

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.

```
/*CREATE PROCEDURE `EXP` ()
BEGIN
SELECT *
FROM EMP_RECORD_TABLE
WHERE EXP >3 ;
END
```

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',

```
/* CREATE DEFINER=`root`@`localhost` PROCEDURE `EXP&ROLE`()
BEGIN
SELECT EMP_ID, FIRST_NAME, LAST_NAME, EXP,
CASE
    WHEN EXP< 2 THEN "JUNIOR DATA SCIENTIST"
    WHEN EXP BETWEEN 2 AND 5 THEN "ASSOCIATE DATA SCIENTIST"
    WHEN EXP BETWEEN 5 AND 10 THEN "SENIOR DATA SCIENTIST"
    WHEN EXP BETWEEN 10 AND 12 THEN "LEAD DATA SCIENTIST"
    WHEN EXP BETWEEN 12 AND 16 THEN "MANAGER"
END AS ROLE
FROM EMP_RECORD_TABLE ORDER BY EXP;
END */
```

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 IDX2
ON EMP_RECORD_TABLE(FIRST_NAME(20));
SHOW INDEXES FROM EMPLOYEE.EMP_RECORD_TABLE;
SELECT *
FROM EMP_RECORD_TABLE
WHERE FIRST_NAME = "ERIC";
```

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
SELECT EMP_ID, FIRST_NAME, SALARY, (0.05*SALARY*EMP_RATING) AS INCREMENT
FROM EMP_RECORD_TABLE;
```

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, SALARY, CONTINENT, COUNTRY,
AVG(SALARY) OVER (PARTITION BY COUNTRY) AVG_SAL_IN_COUNTRY,
AVG(SALARY) OVER (PARTITION BY CONTINENT) AVG_SAL_IN_CONTINENT,
COUNT(*) OVER (PARTITION BY COUNTRY) AVG_SAL_IN_COUNTRY,
COUNT(*) OVER (PARTITION BY COUNTRY) AVG_SAL_IN_CONTINENT
FROM EMP_RECORD_TABLE;
```

THANK YOU ALL

EMAIL ADDRESS

akshaymoodi78@gmail.com