**Objective: Build a PostgreSQL database to manage employee data for a company.**

**Database Design:**

* Design tables to store information about employees, departments, job titles, and salaries.
* Ensure you have the necessary relationships between tables (e.g., each employee belongs to a department).

**Tables and Relationships:**

* Create an employees table with fields like employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, and department\_id.
* Create a departments table with fields like department\_id, department\_name, and manager\_id.
* Create a jobs table with fields like job\_id, job\_title, min\_salary, and max\_salary.
* Ensure you have foreign key constraints where necessary (e.g., department\_id, job\_id in the employees table should reference the departments and jobs table).

**Data Insertion:**

Insert sample data into the tables (e.g., 12 employees, 5 departments, and 7 job titles).

**Queries:**

1. **Creating departments table and inserting the values.**

CREATE TABLE departments(

department\_id INT PRIMARY KEY,

department\_name VARCHAR (100),

manager\_id INT

);

INSERT INTO departments

(department\_id, department\_name, manager\_id)

VALUES

(1, 'HR', 200), (2, 'IT', 201), (3, 'FINANCE', 204), (4, 'MANAGEMENT', 205), (5, 'ANALYST', 203);

1. **Creating jobs table and inserting the values.**

CREATE TABLE jobs(

job\_id INT PRIMARY KEY,

job\_title VARCHAR (100),

min\_salary VARCHAR (100),

max\_salary VARCHAR (100)

);

INSERT INTO jobs (job\_id, job\_title, min\_salary, max\_salary)

VALUES

(400, 'SOFTWARE ENGINEER', '50000', '600000'),

(401, 'DATA ANALYST', '50000', '600000'),

(402, 'SOFTWARE DEVELOPER', '50000', '600000'),

(403, 'JUNIOR CONSULTANT', '50000', '600000'),

(404, 'SENIOR HR', '50000', '600000'),

(405, 'SENIOR CONSULTANT', '50000', '600000'),

(406, 'HR TRAINEE', '50000', '600000');

1. **Creating employees table and inserting the values.**

CREATE TABLE employees(

employee\_id serial primary key,

first\_name VARCHAR (100),

last\_name VARCHAR (100),

email VARCHAR (100) UNIQUE,

phone\_number VARCHAR,

hire\_date Date,

job\_id int,

salary decimal (8,2),

department\_id int,

FOREIGN KEY(department\_id) REFERENCES departments(department\_id),

FOREIGN KEY(job\_id) REFERENCES jobs(job\_id)

);

INSERT INTO employees

(first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, department\_id)

VALUES

('Gaurav', 'Rana', 'gauravranaa.2001@gmail.com', '9821236184', '2024-09-01', 402, 200000, 2),

('Shivani', 'Shah', 'shivani.2001@gmail.com', '8321236184', '2023-09-01', 402, 300000, 2),

('Nandini', 'Mehandiratta', 'nandini.2001@gmail.com', '9521236184', '2024-05-01', 406, 300000, 1),

('Lokesh', 'yadav', 'lokyadav.2001@gmail.com', '8621236184', '2024-04-01', 403, 100000, 3),

('Priyanka', 'Nigam', 'pnigam.2001@gmail.com', '9825236184', '2024-03-01', 404, 600000, 1),

('Ashish', 'Gupta', 'ashishgupta001@gmail.com', '7865236184', '2024-02-01', 405, 500000, 3),

('Aaditya', 'Singh', 'aadityasingh01@gmail.com', '9834236184', '2024-02-01', 400, 200000, 4),

('Devansh', 'Garg', 'devanshgarg001@gmail.com', '9864236184', '2024-01-01', 401, 100000, 3),

('Simran', 'Kalra', 'simrankalra01@gmail.com', '8341236184', '2024-4-04', 401, 100000, 5),

('Dhruv', 'Nagpal', 'dhruvranaa.2001@gmail.com', '9344236184', '2024-05-06', 406, 400000, 5),

('Shubham', 'Kumar', 'shubhama.2001@gmail.com', '7856436184', '2024-06-07', 404, 500000, 4),

('Anchal', 'Sehgal', 'sehgalanchal@gmail.com', '9834636184', '2024-07-12', 403, 70000, 4);

**WRITE SQL QUERIES OF THE FOLLOWING :**

**1. Retrieve a list of employees in a specific department.**

SELECT e.\*, j.job\_id, j.job\_title, d.department\_name

FROM employees e

JOIN departments d ON d.department\_id=e.department\_id

JOIN jobs j ON j.job\_id=e.job\_id

WHERE

d.department\_name='FINANCE';

**2. Find the highest-paid employees and the departments they belong to.**

SELECT

e.department\_id, d.department\_name, CONCAT\_WS(' ', e.first\_name, e.last\_name) AS employee\_name,

MAX(salary) AS max\_salary

FROM employees e

JOIN departments d ON d.department\_id=e.department\_id

WHERE

salary = (SELECT MAX(salary) FROM employees WHERE department\_id=e.department\_id)

GROUP BY e.department\_id, d.department\_name, e.first\_name, e.last\_name;

**3. Calculate the average salary by department.**

SELECT

e.department\_id, d.department\_name, AVG(salary) AS avg\_salary

FROM employees e

JOIN departments d ON d.department\_id=e.department\_id

GROUP BY e.department\_id, d.department\_name;

**4. List employees who were hired in the last year.**

SELECT

e.employee\_id, e.first\_name, hire\_date

FROM employees e

WHERE hire\_date BETWEEN '2023-01-01' AND '2023-12-31';

**5. Create a stored procedure to promote employees by increasing their salary by a percentage.**

CREATE OR REPLACE PROCEDURE update\_emp\_salary

(

increment\_percentage INT,

p\_department\_id INT

)

LANGUAGE plpgsql

AS $$

BEGIN

UPDATE employees

SET salary = salary + ((salary/100)\*increment\_percentage)

WHERE department\_id = p\_department\_id;

END;

$$;

CALL update\_emp\_salary(10, 2);