Task 2 Project: Employee Payroll Management System (PostgreSQL)

Objective: Design and implement an employee payroll system to store, manage, and analyze salary data.

Database Setup: Create a database named payroll_database. Create a table employees with columns: EMPLOYEE_ID (integer), NAME (text), DEPARTMENT (text), EMAIL (text), PHONE_NO (numeric), JOINING_DATE (date), SALARY (numeric), BONUS (numeric), TAX_PERCENTAGE (numeric).

Data Entry: Insert 10 sample employee records.

```
CREATE TABLE
payroll_database=#
payroll_database=# -- See the structure (DDL)
payroll_database=# \d employees
                Table "public.employees"
     Column     |     Type      | Collation | Nullable | Default
----------------+---------------+-----------+----------+---------
 employee_id    | integer       |           | not null |
 name           | text          |           | not null |
 department     | text          |           | not null |
 email          | text          |           |          |
 phone_no       | numeric(15,0) |           |          |
 joining_date   | date          |           | not null |
 salary         | numeric(12,2) |           | not null |
 bonus          | numeric(12,2) |           | not null |
 tax_percentage | numeric(5,2)  |           | not null |
Indexes:
    "employees_pkey" PRIMARY KEY, btree (employee_id)
    "employees_email_key" UNIQUE CONSTRAINT, btree (email)


payroll_database=# INSERT INTO employees VALUES
payroll_database=#   (1,'Alice Johnson','Sales','alice@acme.com',  9198765432l0,'2025-05-12', 85000,  5000, 15.00),
payroll_database=#   (2,'Bob Smith','Engineering','bob@acme.com',  919812345678,'2024-11-03',120000, 10000, 22.00),
payroll_database=#   (3,'Carol Lee','HR','carol@acme.com',        9198555123l5,'2025-02-20', 70000,  3000, 12.00),
payroll_database=#   (4,'Dan Brown','Sales','dan@acme.com',        919844433322,'2024-06-10', 95000,  7000, 18.00),
payroll_database=#   (5,'Eva Green','Marketing','eva@acme.com',    919800011122,'2025-07-01', 78000,  4000, 10.00),
payroll_database=#   (6,'Frank Yang','Engineering','frank@acme.com',919833377799,'2023-12-15',120000, 12000, 25.00),
payroll_database=#   (7,'Grace Kim','Finance','grace@acme.com',    919877766655,'2025-03-05', 95000,  5000, 20.00),
payroll_database=#   (8,'Hank P','Sales','hank@acme.com',          919822233344,'2024-12-22', 88000,  6000, 15.00),
payroll_database=#   (9,'Ivy Fox','Engineering','ivy@acme.com',    919800099988,'2025-01-28', 95000,  5000, 22.00),
payroll_database=#   (10,'Jon Doe','Marketing','jon@acme.com',     919866677788,'2024-09-09', 70000,  3000, 10.00);
INSERT 0 10
payroll_database=#
payroll_database=# -- Verify data
payroll_database=# SELECT * FROM employees ORDER BY employee_id;
 employee_id |     name      | department  |      email      |   phone_no   | joining_date |  salary   |  bonus   | tax_percentage
-------------+---------------+-------------+-----------------+--------------+--------------+-----------+----------+----------------
           1 | Alice Johnson | Sales       | alice@acme.com  | 919876543210 | 2025-05-12   |  85000.00 |  5000.00 |          15.00
           2 | Bob Smith     | Engineering | bob@acme.com    | 919812345678 | 2024-11-03   | 120000.00 | 10000.00 |          22.00
           3 | Carol Lee     | HR          | carol@acme.com  | 919855512345 | 2025-02-20   |  70000.00 |  3000.00 |          12.00
           4 | Dan Brown     | Sales       | dan@acme.com    | 919844433322 | 2024-06-10   |  95000.00 |  7000.00 |          18.00
           5 | Eva Green     | Marketing   | eva@acme.com    | 919800011122 | 2025-07-01   |  78000.00 |  4000.00 |          10.00
           6 | Frank Yang    | Engineering | frank@acme.com  | 919833377799 | 2023-12-15   | 120000.00 | 12000.00 |          25.00
           7 | Grace Kim     | Finance     | grace@acme.com  | 919877766655 | 2025-03-05   |  95000.00 |  5000.00 |          20.00
           8 | Hank P        | Sales       | hank@acme.com   | 919822233344 | 2024-12-22   |  88000.00 |  6000.00 |          15.00
           9 | Ivy Fox       | Engineering | ivy@acme.com    | 919800099988 | 2025-01-28   |  95000.00 |  5000.00 |          22.00
          10 | Jon Doe       | Marketing   | jon@acme.com    | 919866677788 | 2024-09-09   |  70000.00 |  3000.00 |          10.00
(10 rows)
```

Payroll Queries: a) Retrieve the list of employees sorted by salary in descending order.

```
payroll_database=# SELECT employee_id, name, department, salary
payroll_database-# FROM employees
payroll_database-# ORDER BY salary DESC, employee_id ASC;
 employee_id |     name      | department  |  salary
-------------+---------------+-------------+-----------
           2 | Bob Smith     | Engineering | 120000.00
           6 | Frank Yang    | Engineering | 120000.00
           4 | Dan Brown     | Sales       |  95000.00
           7 | Grace Kim     | Finance     |  95000.00
           9 | Ivy Fox       | Engineering |  95000.00
           8 | Hank P        | Sales       |  88000.00
           1 | Alice Johnson | Sales       |  85000.00
           5 | Eva Green     | Marketing   |  78000.00
           3 | Carol Lee     | HR          |  70000.00
          10 | Jon Doe       | Marketing   |  70000.00
(10 rows)


payroll_database=#
```

b) Find employees with a total compensation (SALARY + BONUS) greater than $100,000.

```
payroll_database=# SELECT employee_id, name, salary, bonus,
payroll_database-#        (salary + bonus) AS total_compensation
payroll_database-# FROM employees
payroll_database-# WHERE (salary + bonus) > 100000
payroll_database-# ORDER BY total_compensation DESC;
 employee_id |    name    |  salary   |  bonus   | total_compensation
-------------+------------+-----------+----------+--------------------
           6 | Frank Yang | 120000.00 | 12000.00 |          132000.00
           2 | Bob Smith  | 120000.00 | 10000.00 |          130000.00
           4 | Dan Brown  |  95000.00 |  7000.00 |          102000.00
(3 rows)
```

c) Update the bonus for employees in the ‘Sales’ department by 10%.

```
payroll_database=# UPDATE employees
payroll_database-# SET bonus = ROUND(bonus * 1.10, 2)
payroll_database-# WHERE department = 'Sales';
UPDATE 3
payroll_database=#
payroll_database=# SELECT employee_id, name, department, bonus
payroll_database-# FROM employees
payroll_database-# WHERE department = 'Sales'
payroll_database-# ORDER BY employee_id;
 employee_id |     name      | department |  bonus
-------------+---------------+------------+---------
           1 | Alice Johnson | Sales      | 5500.00
           4 | Dan Brown     | Sales      | 7700.00
           8 | Hank P        | Sales      | 6600.00
(3 rows)
```

d) Calculate the net salary after deducting tax for all employees.

```
payroll_database=# SELECT employee_id, name, salary, bonus, tax_percentage,
payroll_database-#        ROUND( (salary + bonus) * (1 - (tax_percentage/100.0)), 2 ) AS n
payroll_database-# FROM employees
payroll_database-# ORDER BY employee_id;
 employee_id |     name      |  salary   |  bonus   | tax_percentage | net_salary_after_ta
-------------+---------------+-----------+----------+----------------+--------------------
           1 | Alice Johnson |  85000.00 |  5500.00 |          15.00 |            76925.0
           2 | Bob Smith     | 120000.00 | 10000.00 |          22.00 |           101400.0
           3 | Carol Lee     |  70000.00 |  3000.00 |          12.00 |            64240.0
           4 | Dan Brown     |  95000.00 |  7700.00 |          18.00 |            84214.0
           5 | Eva Green     |  78000.00 |  4000.00 |          10.00 |            73800.0
           6 | Frank Yang    | 120000.00 | 12000.00 |          25.00 |            99000.0
           7 | Grace Kim     |  95000.00 |  5000.00 |          20.00 |            80000.0
           8 | Hank P        |  88000.00 |  6600.00 |          15.00 |            80410.0
           9 | Ivy Fox       |  95000.00 |  5000.00 |          22.00 |            78000.0
          10 | Jon Doe       |  70000.00 |  3000.00 |          10.00 |            65700.0
(10 rows)
```

e) Retrieve the average, minimum, and maximum salary per department.

```
payroll_database=# SELECT department,
payroll_database-#        ROUND(AVG(salary),2) AS avg_salary,
payroll_database-#        MIN(salary)          AS min_salary,
payroll_database-#        MAX(salary)          AS max_salary
payroll_database-# FROM employees
payroll_database-# GROUP BY department
payroll_database-# ORDER BY department;
 department  | avg_salary | min_salary | max_salary
-------------+------------+------------+------------
 Engineering |  111666.67 |   95000.00 |  120000.00
 Finance     |   95000.00 |   95000.00 |   95000.00
 HR          |   70000.00 |   70000.00 |   70000.00
 Marketing   |   74000.00 |   70000.00 |   78000.00
 Sales       |   89333.33 |   85000.00 |   95000.00
(5 rows)
```

Advanced Queries: a) Retrieve employees who joined in the last 6 months.

```
payroll_database=# SELECT employee_id, name, joining_date
payroll_database-# FROM employees
payroll_database-# WHERE joining_date >= (CURRENT_DATE - INTERVAL '6 months')
payroll_database-# ORDER BY joining_date DESC;
 employee_id |     name      | joining_date
-------------+---------------+--------------
           5 | Eva Green     | 2025-07-01
           1 | Alice Johnson | 2025-05-12
           7 | Grace Kim     | 2025-03-05
(3 rows)
```

b) Group employees by department and count how many employees each has.

```
payroll_database=# SELECT department, COUNT(*) AS employee_count
payroll_database-# FROM employees
payroll_database-# GROUP BY department
payroll_database-# ORDER BY employee_count DESC, department;
 department  | employee_count
-------------+----------------
 Engineering |              3
 Sales       |              3
 Marketing   |              2
 Finance     |              1
 HR          |              1
(5 rows)
```

c) Find the department with the highest average salary

```
payroll_database=# WITH dept_avgs AS (
payroll_database(#   SELECT department, AVG(salary) AS avg_salary
payroll_database(#   FROM employees
payroll_database(#   GROUP BY department
payroll_database(# )
payroll_database-# SELECT department, ROUND(avg_salary,2) AS avg_salary
payroll_database-# FROM dept_avgs
payroll_database-# ORDER BY avg_salary DESC
payroll_database-# LIMIT 1;
 department  | avg_salary
-------------+------------
 Engineering |  111666.67
(1 row)
```

d) Identify employees who have the same salary as at least one other employee.

```
payroll_database=# SELECT e1.employee_id, e1.name, e1.salary
payroll_database-# FROM employees e1
payroll_database-# JOIN (
payroll_database(#   SELECT salary
payroll_database(#   FROM employees
payroll_database(#   GROUP BY salary
payroll_database(#   HAVING COUNT(*) >= 2
payroll_database(# ) dup ON dup.salary = e1.salary
payroll_database-# ORDER BY e1.salary DESC, e1.employee_id;
 employee_id |    name    |  salary
-------------+------------+----------
           2 | Bob Smith  | 120000.00
           6 | Frank Yang | 120000.00
           4 | Dan Brown  |  95000.00
           7 | Grace Kim  |  95000.00
           9 | Ivy Fox    |  95000.00
           3 | Carol Lee  |  70000.00
          10 | Jon Doe    |  70000.00
(7 rows)
```