PostgreSQL Practice Questions & Corrected Answers

1. Input Data Table

```
CREATE TABLE employees (
    emp_id SERIAL PRIMARY KEY,
    emp_name VARCHAR(50),
    department VARCHAR(50),
    salary NUMERIC(10, 2),
    joining_date DATE
);

INSERT INTO employees (emp_name, department, salary, joining_date) VALUES
('Alice', 'IT', 75000, '2020-06-15'),
('Bob', 'HR', 50000, '2019-07-20'),
('Charlie', 'IT', 90000, '2018-04-10'),
('David', 'Finance', 65000, '2021-08-01'),
('Emma', 'Finance', 72000, '2022-01-25'),
('Frank', 'HR', 48000, '2021-05-30');
```

2. Tricky Questions & Corrected Answers

1. Fetch Data

- Retrieve the details of the second highest-paid employee.
- SELECT * FROM employees ORDER BY salary DESC LIMIT 1 OFFSET 1;
- Retrieve employees who joined before the most recent employee.
- SELECT * FROM employees ORDER BY joining_date DESC LIMIT 1 OFFSET 1;

2. ALTER COLUMN

- Allow NULL values in salary column.
- ALTER TABLE employees ALTER COLUMN salary DROP NOT NULL;
- Increase length of department column.
- ALTER TABLE employees ALTER COLUMN department TYPE VARCHAR(100);

3. UPDATE

- Increase salary of IT employees by 10%.
- UPDATE employees SET salary = salary + (salary * 10 / 100) WHERE department = 'IT';
- Swap salaries of employees in HR and Finance.
- ALTER TABLE employees ADD COLUMN temp_sal INT;

- UPDATE employees SET temp_sal = salary WHERE department = 'HR';
- UPDATE employees SET salary = (SELECT salary FROM employees WHERE department = 'Finance' LIMIT 1) WHERE department = 'HR';
- UPDATE employees SET salary = (SELECT temp_sal FROM employees WHERE department = 'HR' LIMIT 1) WHERE department = 'Finance';
- ALTER TABLE employees DROP COLUMN temp_sal;

4. DELETE

- Delete the employee with the lowest salary while keeping at least one employee per department.
- DELETE FROM employees WHERE emp_id IN (SELECT emp_id FROM employees ORDER BY salary LIMIT 1);

5. SELECT DISTINCT

- Find all unique department names.
- SELECT DISTINCT department FROM employees;
- Count unique salaries.
- SELECT COUNT(DISTINCT salary) FROM employees;

6. WHERE Clause

- Retrieve employees whose names start with 'A' or end with 'e'.
- SELECT * FROM employees WHERE emp_name LIKE 'A%' OR emp_name LIKE '%e';
- Get employees with salary > 50000 but not in IT.
- SELECT * FROM employees WHERE salary > 50000 AND department <> 'IT';

7. ORDER BY

- Retrieve employees in ascending order of salary but descending order of joining date.
- SELECT * FROM employees ORDER BY salary, joining_date DESC;

8. LIMIT

- Fetch the top 3 highest-paid employees.
- SELECT * FROM employees ORDER BY salary DESC LIMIT 3;

9. MIN & MAX

- Find the department with the highest average salary.
- SELECT department, AVG(salary) AS average_salary FROM employees GROUP BY department ORDER BY average_salary DESC LIMIT 1;
- Retrieve the most recently joined employee.
- SELECT emp_name FROM employees ORDER BY joining_date DESC LIMIT 1;

10. COUNT

- Count employees in each department.
- SELECT department, COUNT(emp_name) AS no_of_employee FROM employees GROUP BY department;

11. AVG & SUM

- Find total and average salary of all employees.
- SELECT AVG(salary) AS 'average salary', SUM(salary) AS 'total salary' FROM employees;
- Find the average salary of employees who joined before 2021.

- SELECT AVG(salary) AS avg_sal FROM employees WHERE joining_date < '2021-01-01';

12. LIKE Clause

- Fetch all employees whose names contain 'a' at least twice.
- SELECT emp_name FROM employees WHERE emp_name LIKE '%a%a%';

13. IN Clause

- Retrieve employees who are in IT or Finance departments.
- SELECT * FROM employees WHERE department IN ('IT', 'Finance');

14. BETWEEN Clause

- Fetch employees whose salary is between 50,000 and 80,000.
- SELECT emp_name, salary FROM employees WHERE salary BETWEEN 50000 AND 80000;

15. AS Clause (Alias)

- Retrieve employee names along with salaries but rename columns as 'Employee' and 'Earnings'.
- SELECT emp_name AS Employee, salary AS Earnings FROM employees;