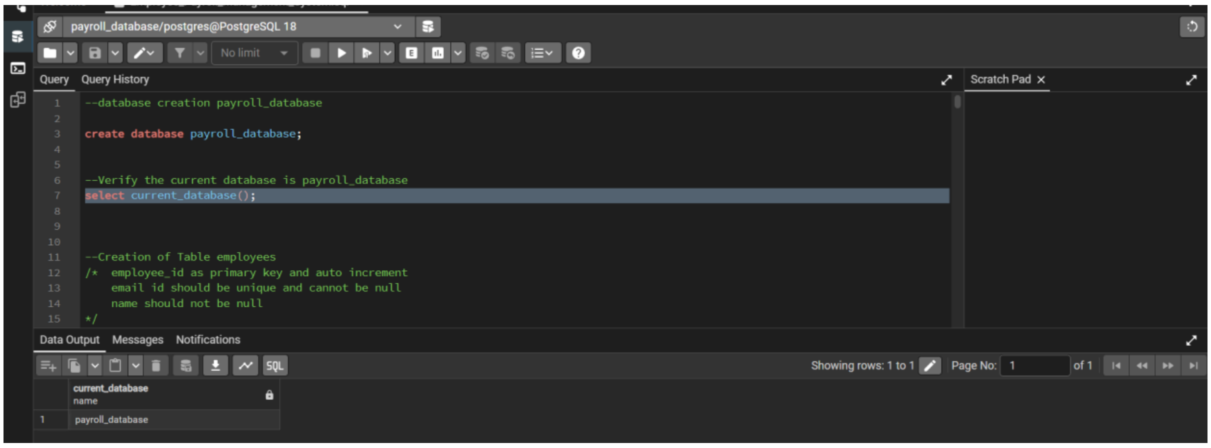
Task 2

Project: Employee Payroll Management System (PostgreSQL)

Objective:

Design and implement an employee payroll system to store, manage, and analyse salary data.

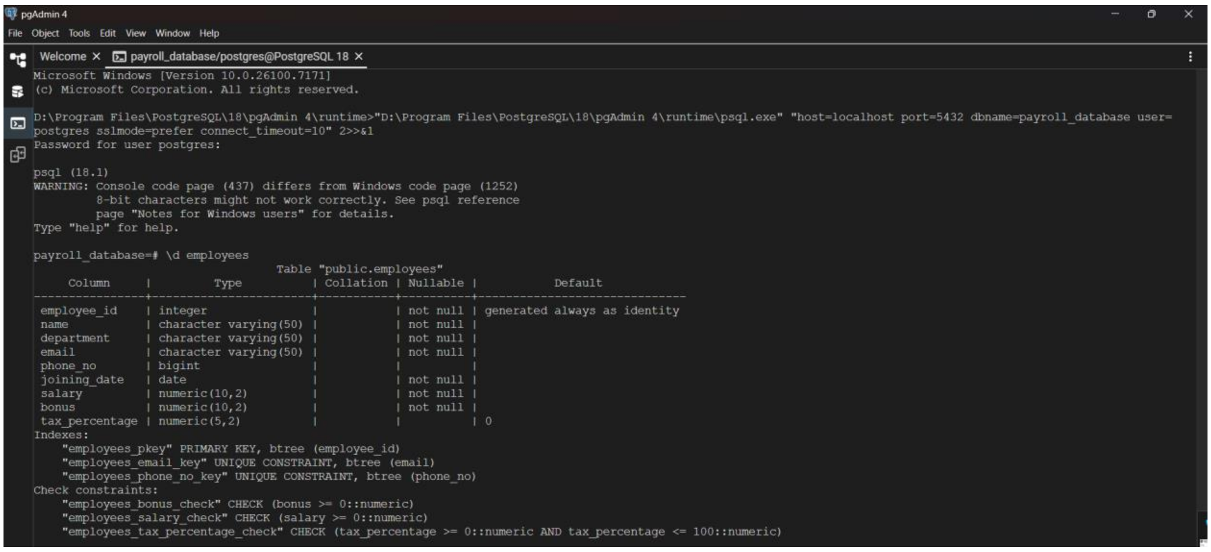
Database created and selected;



```
--database creation payroll_database

create database payroll_database;


--Verify the current database is payroll_database
select current_database();



--Creation of Table employees
/*  employee_id as primary key and auto increment
    email id should be unique and cannot be null
    name should not be null
*/
```

```
current_database
name
1   payroll_database
```

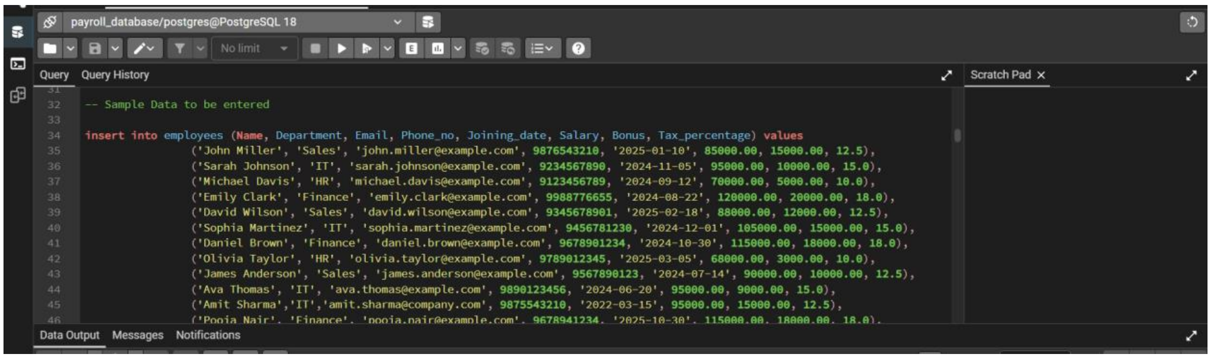
Table created:



```
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

D:\Program Files\PostgreSQL\18\pgAdmin 4\runtime>"D:\Program Files\PostgreSQL\18\pgAdmin 4\runtime\psql.exe" "host=localhost port=5432 dbname=payroll_database user=
postgres sslmode=prefer connect_timeout=10" 2>>&1
Password for user postgres:

psql (18.1)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

payroll_database=# \d employees
                                    Table "public.employees"
     Column     |         Type          | Collation | Nullable |           Default
----------------+-----------------------+-----------+----------+------------------------------
 employee_id    | integer               |           | not null | generated always as identity
 name           | character varying(50) |           | not null |
 department     | character varying(50) |           | not null |
 email          | character varying(50) |           | not null |
 phone_no       | bigint                |           |          |
 joining_date   | date                  |           | not null |
 salary         | numeric(10,2)         |           | not null |
 bonus          | numeric(10,2)         |           | not null |
 tax_percentage | numeric(5,2)          |           |          | 0
Indexes:
    "employees_pkey" PRIMARY KEY, btree (employee_id)
    "employees_email_key" UNIQUE CONSTRAINT, btree (email)
    "employees_phone_no_key" UNIQUE CONSTRAINT, btree (phone_no)
Check constraints:
    "employees_bonus_check" CHECK (bonus >= 0::numeric)
    "employees_salary_check" CHECK (salary >= 0::numeric)
    "employees_tax_percentage_check" CHECK (tax_percentage >= 0::numeric AND tax_percentage <= 100::numeric)
```

Data Entry:

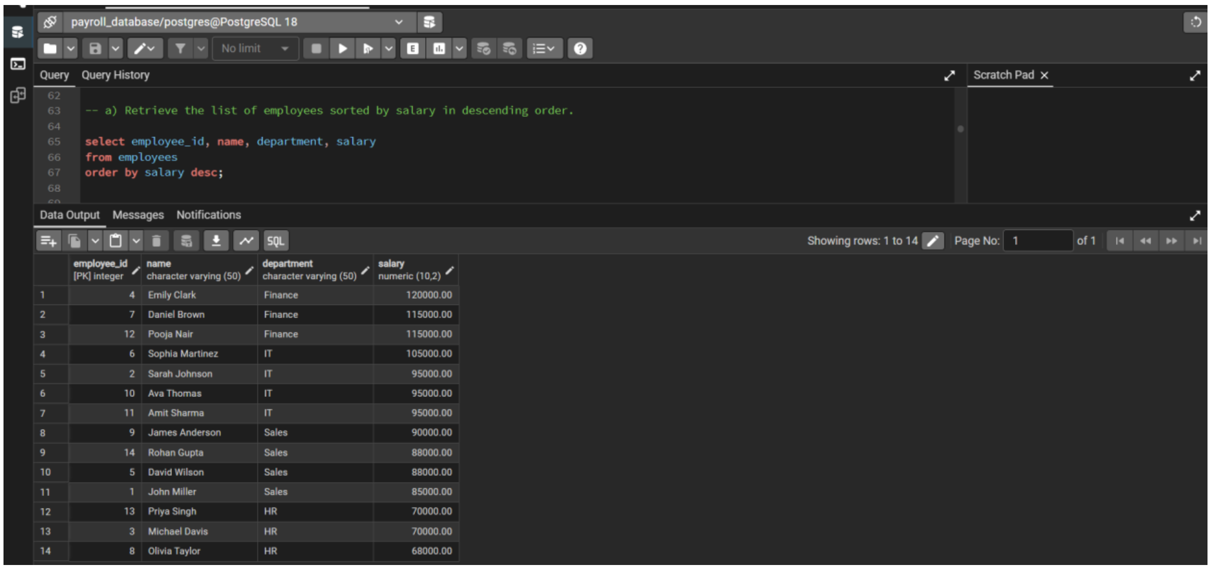
Insert 10 sample employee records.



```
-- Sample Data to be entered

insert into employees (Name, Department, Email, Phone_no, Joining_date, Salary, Bonus, Tax_percentage) values
                ('John Miller', 'Sales', 'john.miller@example.com', 9876543210, '2025-01-10', 85000.00, 15000.00, 12.5),
                ('Sarah Johnson', 'IT', 'sarah.johnson@example.com', 9234567890, '2024-11-05', 95000.00, 10000.00, 15.0),
                ('Michael Davis', 'HR', 'michael.davis@example.com', 9123456789, '2024-09-12', 70000.00, 5000.00, 10.0),
                ('Emily Clark', 'Finance', 'emily.clark@example.com', 9988776655, '2024-08-22', 120000.00, 20000.00, 18.0),
                ('David Wilson', 'Sales', 'david.wilson@example.com', 9345678901, '2025-02-18', 88000.00, 12000.00, 12.5),
                ('Sophia Martinez', 'IT', 'sophia.martinez@example.com', 9456781230, '2024-12-01', 105000.00, 15000.00, 15.0),
                ('Daniel Brown', 'Finance', 'daniel.brown@example.com', 9678901234, '2024-10-30', 115000.00, 18000.00, 18.0),
                ('Olivia Taylor', 'HR', 'olivia.taylor@example.com', 9789012345, '2025-03-05', 68000.00, 3000.00, 10.0),
                ('James Anderson', 'Sales', 'james.anderson@example.com', 9567890123, '2024-07-14', 90000.00, 10000.00, 12.5),
                ('Ava Thomas', 'IT', 'ava.thomas@example.com', 9890123456, '2024-06-20', 95000.00, 9000.00, 15.0),
                ('Amit Sharma','IT','amit.sharma@company.com', 9875543210, '2022-03-15', 95000.00, 15000.00, 12.5),
                ('Pooja Nair', 'Finance', 'pooja.nair@example.com', 9678941234, '2025-10-30', 115000.00, 18000.00, 18.0),
```

Payroll Queries:

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



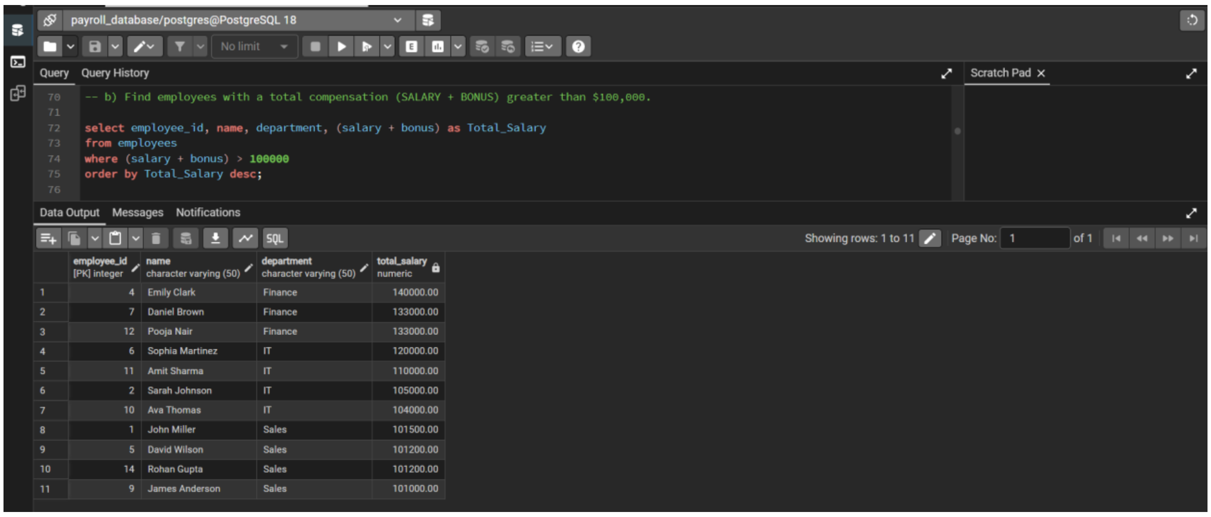
payroll_database/postgres@PostgreSQL 18

```sql
-- a) Retrieve the list of employees sorted by salary in descending order.

select employee_id, name, department, salary
from employees
order by salary desc;
```

Showing rows: 1 to 14 Page No: 1 of 1

	employee_id [PK] integer	name character varying (50)	department character varying (50)	salary numeric (10,2)
1	4	Emily Clark	Finance	120000.00
2	7	Daniel Brown	Finance	115000.00
3	12	Pooja Nair	Finance	115000.00
4	6	Sophia Martinez	IT	105000.00
5	2	Sarah Johnson	IT	95000.00
6	10	Ava Thomas	IT	95000.00
7	11	Amit Sharma	IT	95000.00
8	9	James Anderson	Sales	90000.00
9	14	Rohan Gupta	Sales	88000.00
10	5	David Wilson	Sales	88000.00
11	1	John Miller	Sales	85000.00
12	13	Priya Singh	HR	70000.00
13	3	Michael Davis	HR	70000.00
14	8	Olivia Taylor	HR	68000.00

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



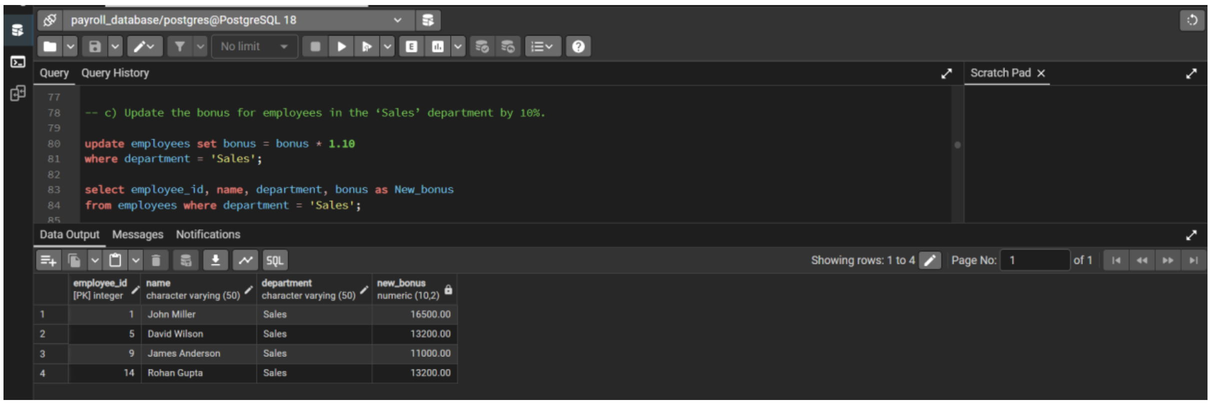
payroll_database/postgres@PostgreSQL 18

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

select employee_id, name, department, (salary + bonus) as Total_Salary
from employees
where (salary + bonus) > 100000
order by Total_Salary desc;
```

Showing rows: 1 to 11 Page No: 1 of 1

	employee_id [PK] integer	name character varying (50)	department character varying (50)	total_salary numeric
1	4	Emily Clark	Finance	140000.00
2	7	Daniel Brown	Finance	133000.00
3	12	Pooja Nair	Finance	133000.00
4	6	Sophia Martinez	IT	120000.00
5	11	Amit Sharma	IT	110000.00
6	2	Sarah Johnson	IT	105000.00
7	10	Ava Thomas	IT	104000.00
8	1	John Miller	Sales	101500.00
9	5	David Wilson	Sales	101200.00
10	14	Rohan Gupta	Sales	101200.00
11	9	James Anderson	Sales	101000.00

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



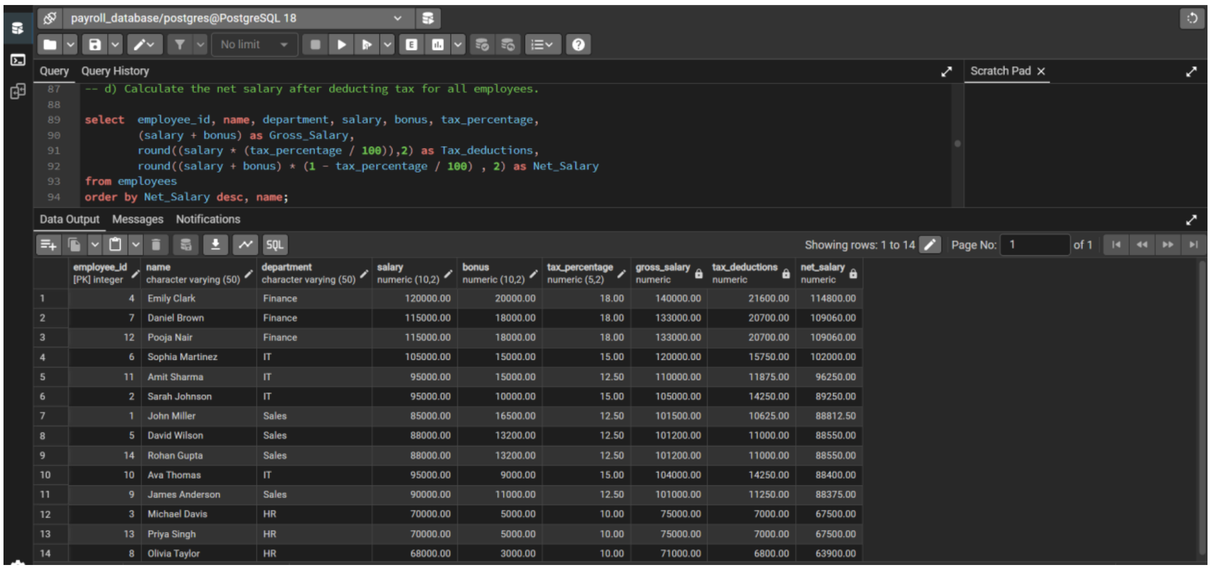
payroll_database/postgres@PostgreSQL 18

```sql
-- c) Update the bonus for employees in the 'Sales' department by 10%.

update employees set bonus = bonus * 1.10
where department = 'Sales';

select employee_id, name, department, bonus as New_bonus
from employees where department = 'Sales';
```

Showing rows: 1 to 4 Page No: 1 of 1

	employee_id [PK] integer	name character varying (50)	department character varying (50)	new_bonus numeric (10,2)
1	1	John Miller	Sales	16500.00
2	5	David Wilson	Sales	13200.00
3	9	James Anderson	Sales	11000.00
4	14	Rohan Gupta	Sales	13200.00

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

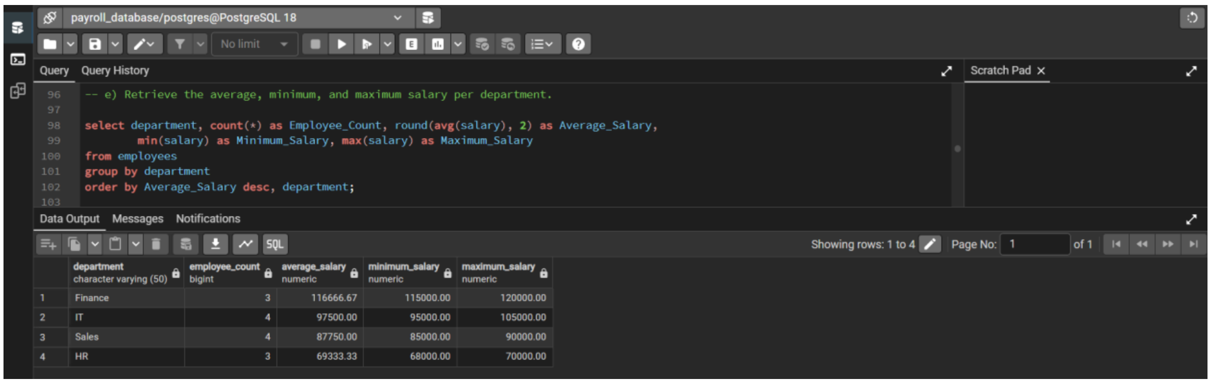


```
-- d) Calculate the net salary after deducting tax for all employees.

select  employee_id, name, department, salary, bonus, tax_percentage,
        (salary + bonus) as Gross_Salary,
        round((salary * (tax_percentage / 100)),2) as Tax_deductions,
        round((salary + bonus) * (1 - tax_percentage / 100) , 2) as Net_Salary
from employees
order by Net_Salary desc, name;
```

Showing rows: 1 to 14 Page No: 1 of 1

	employee_id [PK] integer	name character varying (50)	department character varying (50)	salary numeric (10,2)	bonus numeric (10,2)	tax_percentage numeric (5,2)	gross_salary numeric	tax_deductions numeric	net_salary numeric
1	4	Emily Clark	Finance	120000.00	20000.00	18.00	140000.00	21600.00	114800.00
2	7	Daniel Brown	Finance	115000.00	18000.00	18.00	133000.00	20700.00	109060.00
3	12	Pooja Nair	Finance	115000.00	18000.00	18.00	133000.00	20700.00	109060.00
4	6	Sophia Martinez	IT	105000.00	15000.00	15.00	120000.00	15750.00	102000.00
5	11	Amit Sharma	IT	95000.00	15000.00	12.50	110000.00	11875.00	96250.00
6	2	Sarah Johnson	IT	95000.00	10000.00	15.00	105000.00	14250.00	89250.00
7	1	John Miller	Sales	85000.00	16500.00	12.50	101500.00	10625.00	88812.50
8	5	David Wilson	Sales	88000.00	13200.00	12.50	101200.00	11000.00	88550.00
9	14	Rohan Gupta	Sales	88000.00	13200.00	12.50	101200.00	11000.00	88550.00
10	10	Ava Thomas	IT	95000.00	9000.00	15.00	104000.00	14250.00	88400.00
11	9	James Anderson	Sales	90000.00	11000.00	12.50	101000.00	11250.00	88375.00
12	3	Michael Davis	HR	70000.00	5000.00	10.00	75000.00	7000.00	67500.00
13	13	Priya Singh	HR	70000.00	5000.00	10.00	75000.00	7000.00	67500.00
14	8	Olivia Taylor	HR	68000.00	3000.00	10.00	71000.00	6800.00	63900.00

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



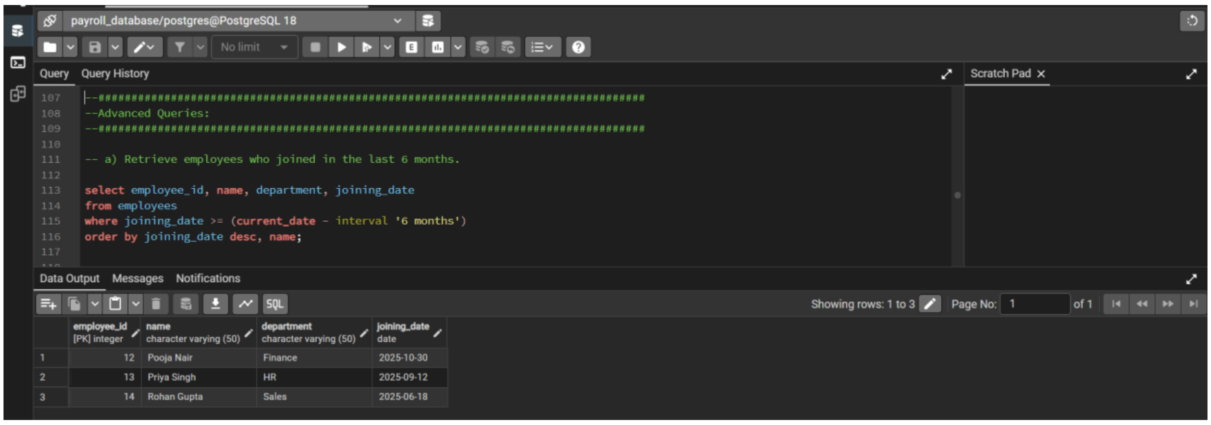
```
-- e) Retrieve the average, minimum, and maximum salary per department.

select department, count(*) as Employee_Count, round(avg(salary), 2) as Average_Salary,
        min(salary) as Minimum_Salary, max(salary) as Maximum_Salary
from employees
group by department
order by Average_Salary desc, department;
```

Showing rows: 1 to 4 Page No: 1 of 1

	department character varying (50)	employee_count bigint	average_salary numeric	minimum_salary numeric	maximum_salary numeric
1	Finance	3	116666.67	115000.00	120000.00
2	IT	4	97500.00	95000.00	105000.00
3	Sales	4	87750.00	85000.00	90000.00
4	HR	3	69333.33	68000.00	70000.00

Advanced Queries:

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

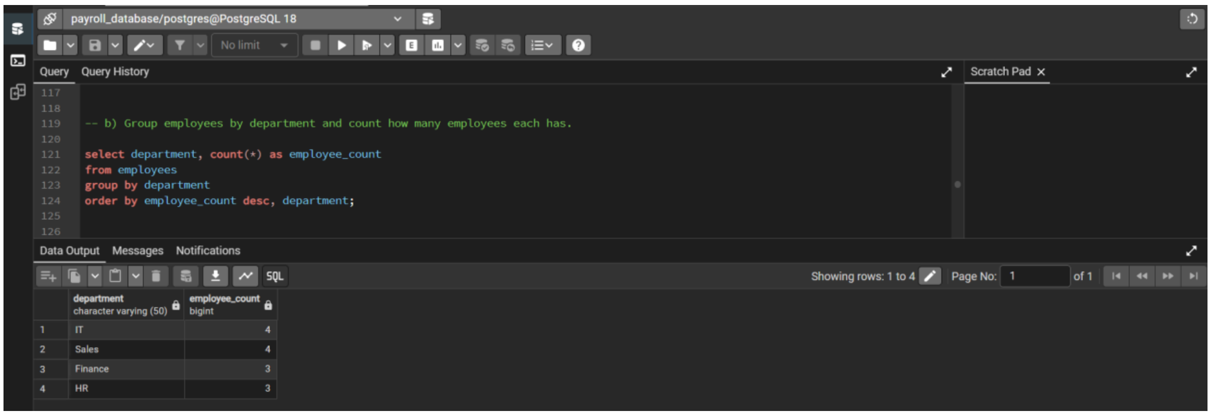


```
--##########################################################################
--Advanced Queries:
--##########################################################################

-- a) Retrieve employees who joined in the last 6 months.

select employee_id, name, department, joining_date
from employees
where joining_date >= (current_date - interval '6 months')
order by joining_date desc, name;
```

Showing rows: 1 to 3 Page No: 1 of 1

	employee_id [PK] integer	name character varying (50)	department character varying (50)	joining_date date
1	12	Pooja Nair	Finance	2025-10-30
2	13	Priya Singh	HR	2025-09-12
3	14	Rohan Gupta	Sales	2025-06-18

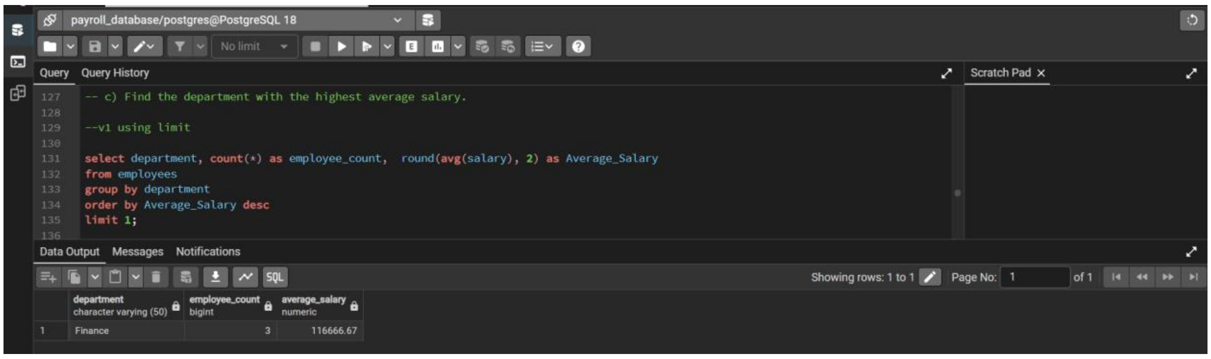
b) Group employees by department and count how many employees each has.



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

select department, count(*) as employee_count
from employees
group by department
order by employee_count desc, department;
```

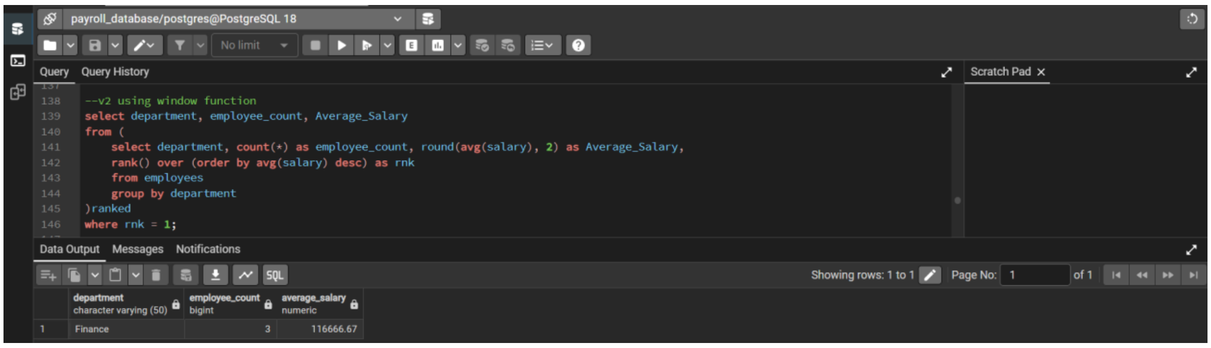
```
  department                employee_count
  character varying (50)    bigint
1 IT                                     4
2 Sales                                  4
3 Finance                                3
4 HR                                     3
```

c) Find the department with the highest average salary.



```
-- c) Find the department with the highest average salary.

--v1 using limit

select department, count(*) as employee_count,  round(avg(salary), 2) as Average_Salary
from employees
group by department
order by Average_Salary desc
limit 1;
```

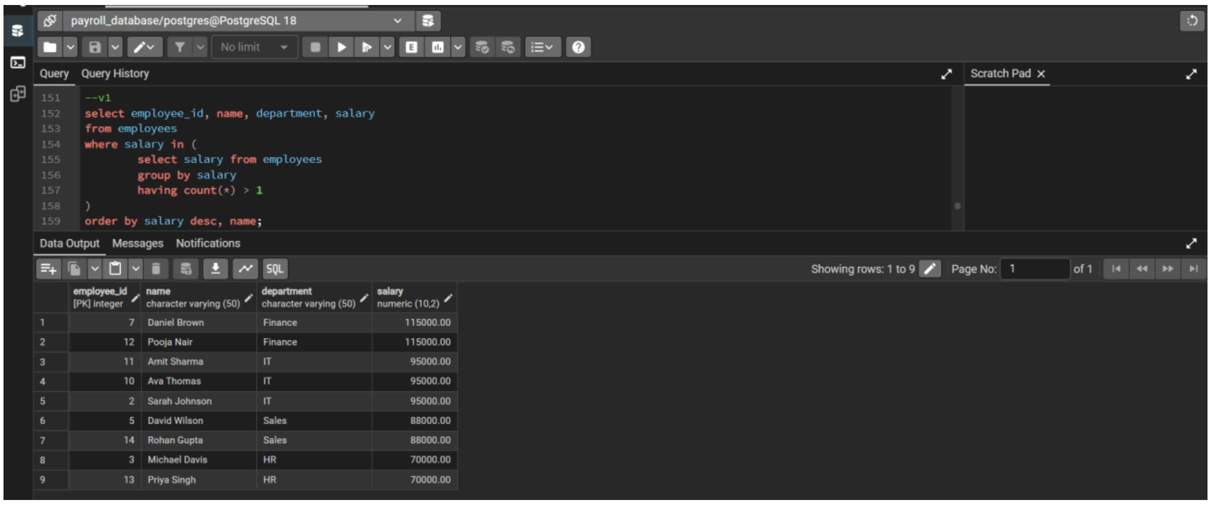
```
  department                employee_count   average_salary
  character varying (50)    bigint           numeric
1 Finance                                3        116666.67
```



```
--v2 using window function
select department, employee_count, Average_Salary
from (
    select department, count(*) as employee_count, round(avg(salary), 2) as Average_Salary,
    rank() over (order by avg(salary) desc) as rnk
    from employees
    group by department
)ranked
where rnk = 1;
```

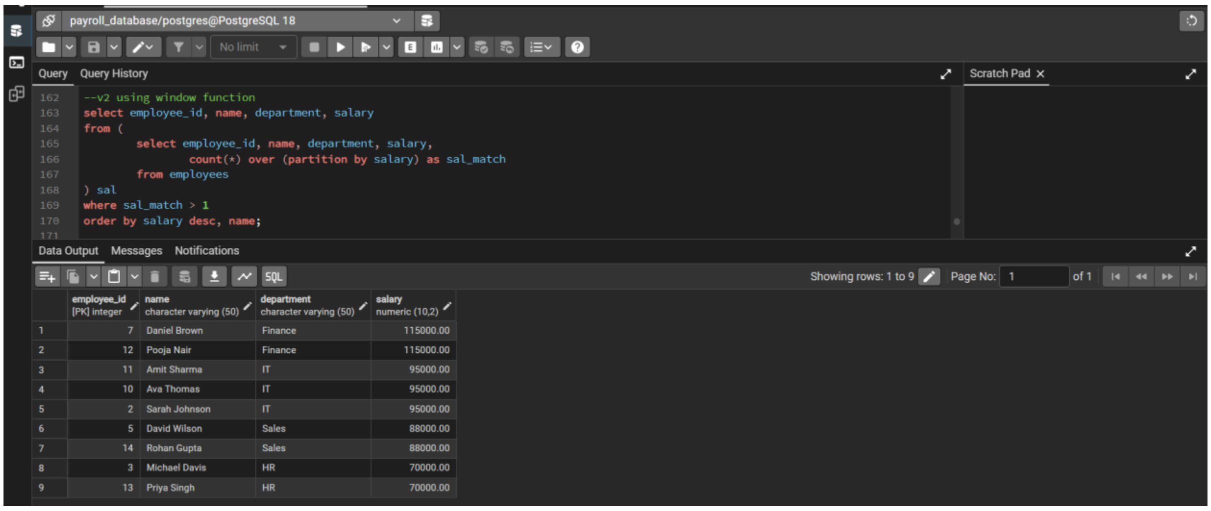
```
  department                employee_count   average_salary
  character varying (50)    bigint           numeric
1 Finance                                3        116666.67
```

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



```
--v1
select employee_id, name, department, salary
from employees
where salary in (
        select salary from employees
        group by salary
        having count(*) > 1
)
order by salary desc, name;
```

```
  employee_id   name                     department               salary
  [PK] integer  character varying (50)   character varying (50)   numeric (10,2)
1            7  Daniel Brown             Finance                      115000.00
2           12  Pooja Nair               Finance                      115000.00
3           11  Amit Sharma              IT                            95000.00
4           10  Ava Thomas               IT                            95000.00
5            2  Sarah Johnson            IT                            95000.00
6            5  David Wilson             Sales                         88000.00
7           14  Rohan Gupta              Sales                         88000.00
8            3  Michael Davis            HR                            70000.00
9           13  Priya Singh              HR                            70000.00
```



```sql
--v2 using window function
select employee_id, name, department, salary
from (
        select employee_id, name, department, salary,
                count(*) over (partition by salary) as sal_match
        from employees
) sal
where sal_match > 1
order by salary desc, name;
```

	employee_id [PK] integer	name character varying (50)	department character varying (50)	salary numeric (10,2)
1	7	Daniel Brown	Finance	115000.00
2	12	Pooja Nair	Finance	115000.00
3	11	Amit Sharma	IT	95000.00
4	10	Ava Thomas	IT	95000.00
5	2	Sarah Johnson	IT	95000.00
6	5	David Wilson	Sales	88000.00
7	14	Rohan Gupta	Sales	88000.00
8	3	Michael Davis	HR	70000.00
9	13	Priya Singh	HR	70000.00