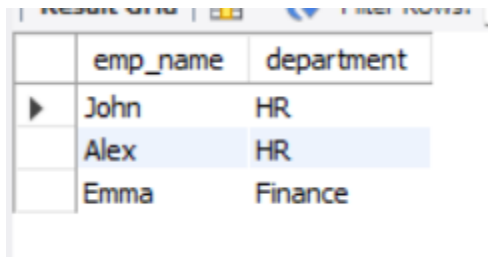


Example 1:

Find all employees who work in the same department as either John or Emma.

```
select emp_name,department
from employees
where department in(
select department from employees where emp_name
in('John','Emma'));
```

OUTPUT:

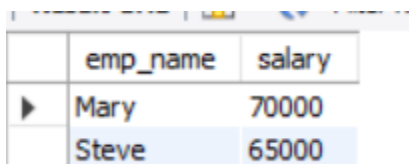
A screenshot of a database query result window. The window title is 'Result Grid'. It shows a table with two columns: 'emp_name' and 'department'. The table contains three rows: John in HR, Alex in HR, and Emma in Finance. The first row is highlighted with a mouse cursor.

	emp_name	department
▶	John	HR
	Alex	HR
	Emma	Finance

Example 2:

Find all employees whose salary equals any salary from the IT department.

```
select emp_name,salary
from employees
where salary= any(
select salary from employees where department='IT' );
```

OUTPUT:

A screenshot of a database query result window. The window title is 'Result Grid'. It shows a table with two columns: 'emp_name' and 'salary'. The table contains two rows: Mary with a salary of 70000 and Steve with a salary of 65000. The first row is highlighted with a mouse cursor.

	emp_name	salary
▶	Mary	70000
	Steve	65000

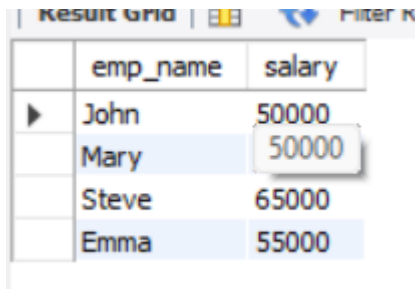
Example 1:

Find employees whose salary is greater than any salary of HR department employees.

```
select emp_name,salary
```

```
from employees
```

```
where salary>any(select salary from employees where  
department='HR');
```

OUTPUT:

	emp_name	salary
▶	John	50000
	Mary	50000
	Steve	65000
	Emma	55000

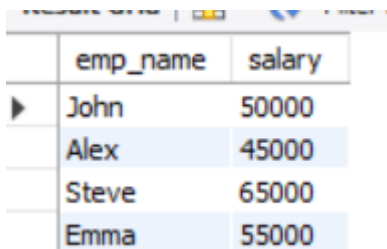
Example 2:

Find employees whose salary is less than any salary in the IT department.

```
select emp_name,salary
```

```
from employees
```

```
where salary<any(select salary from employees where  
department='IT');
```

OUTPUT:

	emp_name	salary
▶	John	50000
	Alex	45000
	Steve	65000
	Emma	55000

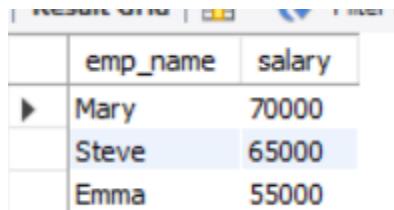
Example 1:

Find employees whose salary is greater than all HR employees.

```
select emp_name,salary
```

```
from employees
```

```
where salary>all(select salary from employees where  
department='HR');
```

OUTPUT:

	emp_name	salary
▶	Mary	70000
	Steve	65000
	Emma	55000

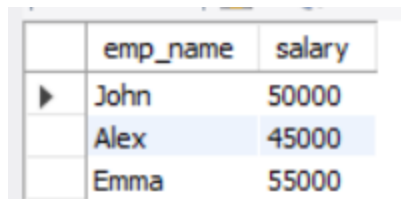
Example 2:

Find employees whose salary is less than all IT employees.

```
select emp_name,salary
```

```
from employees
```

```
where salary<all(select salary from employees where  
department='IT');
```

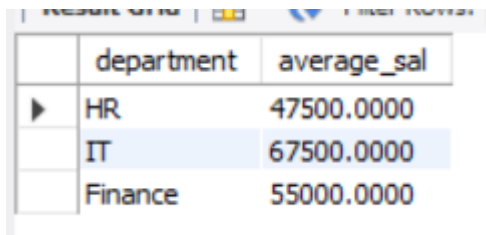
OUTPUT:

	emp_name	salary
▶	John	50000
	Alex	45000
	Emma	55000

Q1. Find employees whose salary is greater than the average salary of any department.

```
select department, avg(salary) as average_sal  
from employees  
group by department  
having avg(salary) > any(select salary from employees);
```

OUTPUT:



A screenshot of a database query result window. It shows a table with two columns: 'department' and 'average_sal'. The table contains three rows: HR with an average salary of 47500.0000, IT with 67500.0000, and Finance with 55000.0000. The IT row is highlighted in blue.

department	average_sal
HR	47500.0000
IT	67500.0000
Finance	55000.0000

Q2. Find departments having employees working in the same department as John or Emma, and show total employees per department.

```
select department, count(*) as total_employees  
from employees  
where department in(  
select department from employees where emp_name  
in('John', 'Emma'))  
group by department  
having count(*) > 1;
```

OUTPUT:

	department	total_employees
▶	HR	2

Q3. Find employees whose salary is greater than any employee in the HR department.

```
select emp_name,salary
```

```
from employees
```

```
where salary>any(select salary from employees where  
department='HR');
```

OUTPUT:

	emp_name	salary
▶	John	50000
	Mary	70000
	Steve	65000
	Emma	55000

Q4. Find employees earning less than any average department salary (based on grouped averages).

```
SELECT emp_name, salary
```

```
FROM employees
```

```
WHERE salary < ANY (
```

```
    SELECT AVG(salary)
```

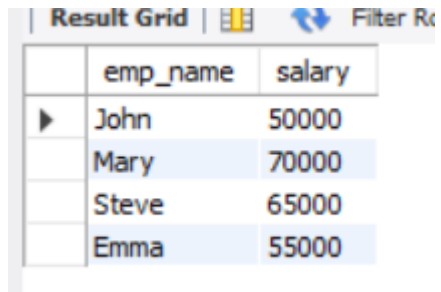
```
    FROM employees
```

```
    GROUP BY department
```

```
)
```

ORDER BY salary DESC;

OUTPUT:



The image shows a screenshot of a database application's 'Result Grid'. The grid has two columns: 'emp_name' and 'salary'. The data is sorted by salary in descending order. The rows are: John (50000), Mary (70000), Steve (65000), and Emma (55000). The rows are alternatingly highlighted with light blue and white backgrounds. The grid is titled 'Result Grid' and has a 'Filter Rows' button.

	emp_name	salary
▶	John	50000
	Mary	70000
	Steve	65000
	Emma	55000