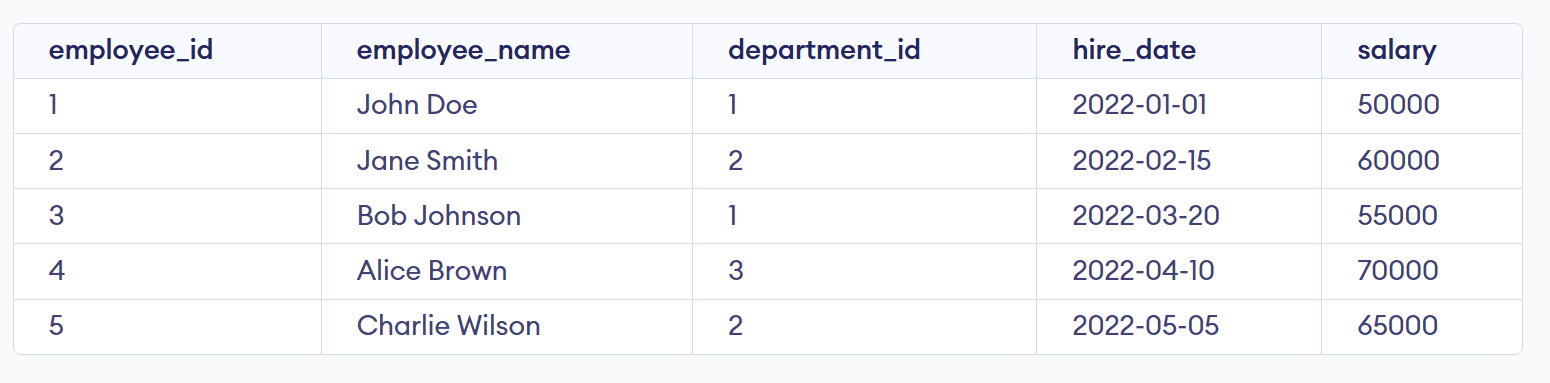
# SQL Project

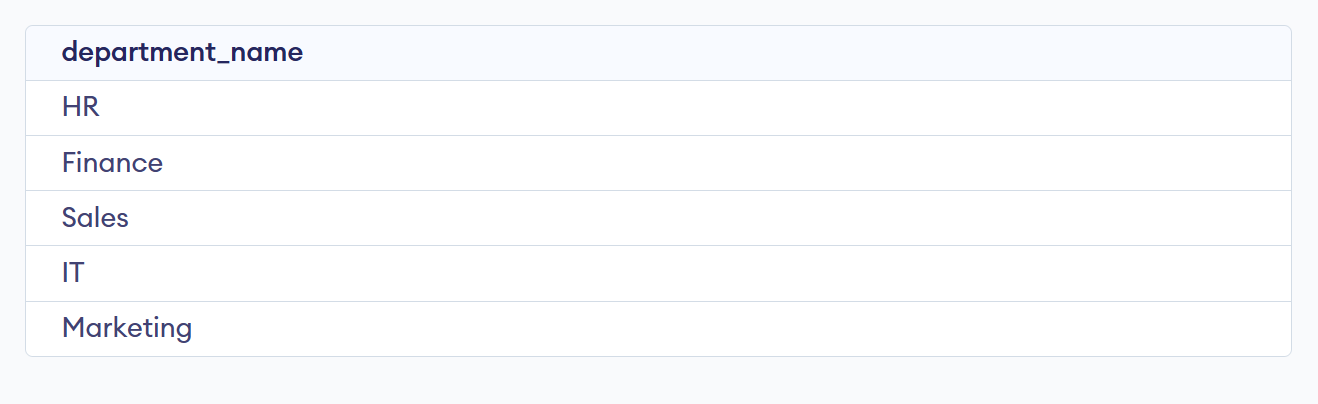
## 1. Select all columns from the 'employee' table:

SELECT \* FROM employee



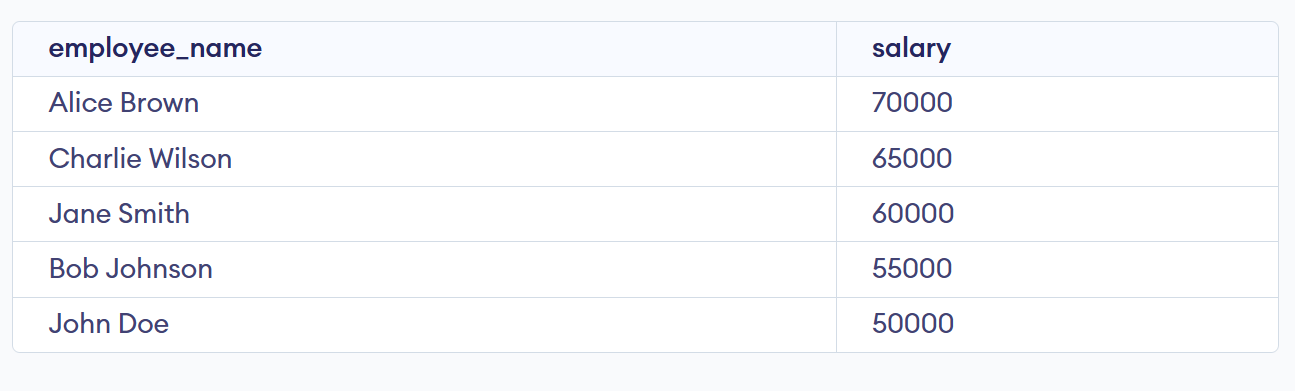
## 2. Select unique department names from the 'department' table:

SELECT DISTINCT department\_name FROM department



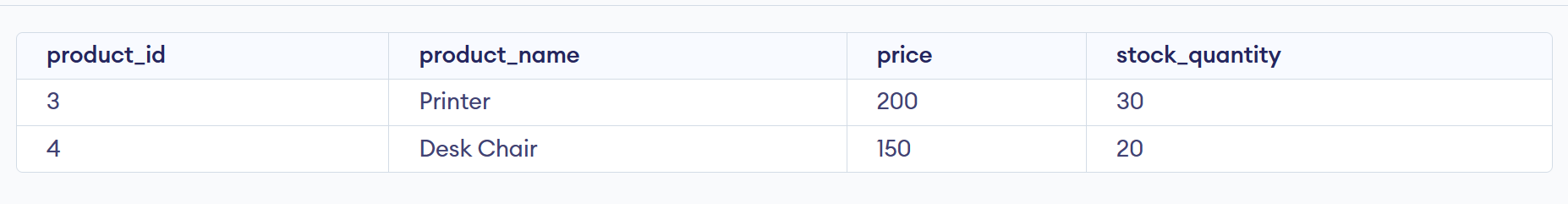
## 3. Select employee names and salaries from the 'employee' table, ordered by salary in descending order:

SELECT employee\_name, salary FROM Employee ORDER BY salary DESC



## 4. Select products with a price between £100 and £500 from the 'product' table:

SELECT \* FROM Product WHERE price BETWEEN 100 AND 500



## 5. Select orders made by employees in the 'IT' department, joining 'purchase' and 'employee' tables:

The correct query to resolve this exercise is the following:

SELECT order\_id, order\_date, employee\_name

FROM purchase

JOIN employee ON employee.employee\_id = purchase.employee\_id

WHERE employee.department\_id = (SELECT department\_id FROM Department WHERE department\_name = "IT");

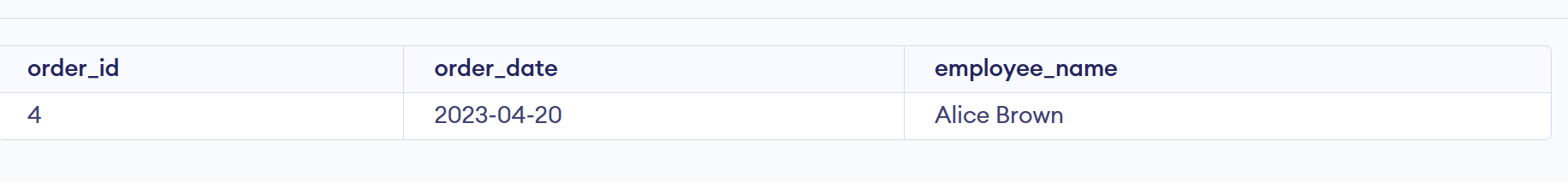
However, this results in no output data, which differs from the expected output shown in the course. The expected output shows the employee’s name Alice Brown, who is in the Sales department. The query to obtain said employee is the following:

SELECT order\_id, order\_date, employee\_name

FROM purchase

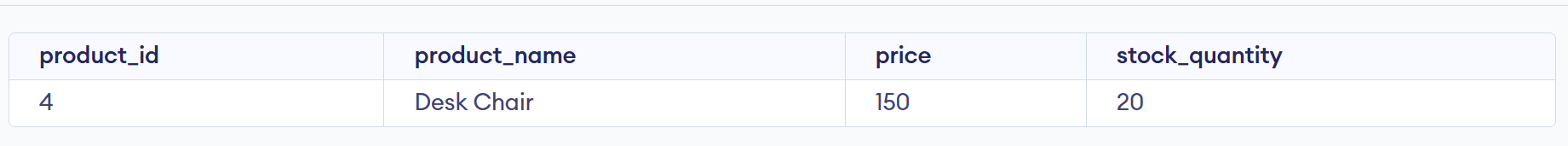
JOIN employee ON employee.employee\_id = purchase.employee\_id

WHERE employee.department\_id = (SELECT department\_id FROM Department WHERE department\_name = "Sales");



## 6. Select products with names containing 'Chair' using a wildcard:

SELECT product\_id, product\_name, price, stock\_quantity FROM product WHERE product\_name LIKE "%Chair%"



## 7. Select orders made by employees with IDs 1, 3, and 5:

SELECT order\_id, employee.employee\_id, product\_id, order\_date, quantity

FROM employee

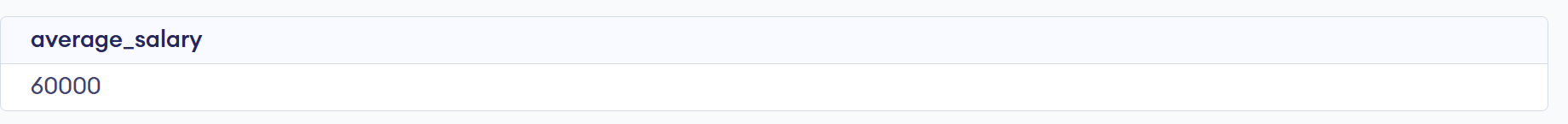
JOIN purchase on employee.employee\_id = purchase.employee\_id

WHERE purchase.employee\_id IN (1, 3, 5)



## 8. Select the average salary of all employees:

SELECT AVG (salary) AS average\_salary from employee

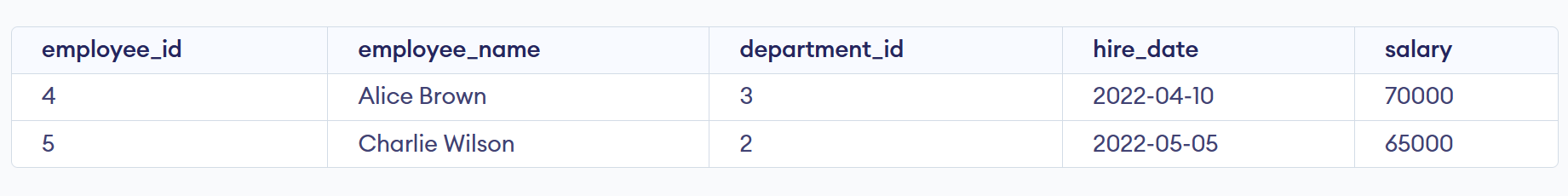


## 9. Select employees hired after '2022-03-01' with a salary greater than £60000:

SELECT employee\_id, employee\_name, department\_id, hire\_date, salary

FROM employee

WHERE hire\_date > 2022-03-01 AND salary > 60000



## 10. Select departments with more than 1 employees, counting the number of employees in each department:

SELECT department.department\_id, department\_name, COUNT (Employee.department\_id) AS num\_employees

FROM department

JOIN employee ON department.department\_id = employee.department\_id

GROUP BY department.department\_id, department\_name

HAVING num\_employees > 1;

