

**1. Wyświetl wszystkich pracowników posortowanych rosnąco po nazwisku, a następnie po imieniu.**

```
SELECT *
FROM Employees
ORDER BY LastName ASC, FirstName ASC;
```

	EmployeeID	FirstName	LastName	DepartmentID	ManagerID	Salary	HireDate
1	7	Katarzyna	Dąbrowska	1	1	130000.00	2021-02-28
2	10	Dorota	Jankowska	2	2	115000.00	2023-01-10
3	8	Paweł	Kamiński	1	7	90000.00	2022-04-01
4	5	Agnieszka	Kowalczyk	3	1	140000.00	2019-11-05
5	9	Łukasz	Kozłowski	3	5	95000.00	2022-06-15
6	6	Michał	Lewandowski	3	5	100000.00	2020-08-12
7	1	Anna	Nowak	1	NULL	250000.00	2018-03-01
8	3	Marta	Wiśniewska	2	2	120000.00	2020-01-20
9	4	Tomasz	Wójcik	2	2	110000.00	2021-05-10
10	2	Piotr	Zieliński	2	1	150000.00	2019-07-15

**2. Dla każdego działu policz liczbę pracowników i wypisz nazwy działów, posortowane malejąco po liczbie pracowników.**

```
SELECT d.DepartmentName, COUNT(e.EmployeeID) AS EmployeeCount
FROM Departments d
JOIN Employees e ON d.DepartmentID = e.DepartmentID
GROUP BY d.DepartmentName
ORDER BY EmployeeCount DESC;
```

	DepartmentName	EmployeeCount
1	IT	4
2	Sales	3
3	HR	3

**3. Oblicz sumę wynagrodzeń w każdym dziale, wypisz tylko te działy, gdzie łączna kwota przekracza 150 000, posortuj malejąco.**

```
SELECT d.DepartmentName, SUM(e.Salary) AS TotalSalary
FROM Departments d
JOIN Employees e ON d.DepartmentID = e.DepartmentID
GROUP BY d.DepartmentName
HAVING SUM(e.Salary) > 150000
ORDER BY TotalSalary DESC;
```

	DepartmentName	TotalSalary
1	IT	495000.00
2	HR	470000.00
3	Sales	335000.00

**4. Lista wszystkich projektów wraz z liczbą przypisanych do nich pracowników (wypisz również projekty bez pracowników).**

```
SELECT p.ProjectName, COUNT(ep.EmployeeID) AS EmployeeCount
FROM Projects p
LEFT JOIN EmployeeProjects ep ON p.ProjectID = ep.ProjectID
GROUP BY p.ProjectName;
```

	ProjectName	EmployeeCount
1	Data Analysis	3
2	Mobile App	3
3	Website Redesign	3

**5. Wyciągnij imiona i nazwiska pracowników, których pensja jest wyższa od średniej pensji wszystkich pracowników (użyj podzapytania).**

```
SELECT FirstName, LastName, Salary
FROM Employees
WHERE Salary > (SELECT AVG(Salary) FROM Employees);
```

	FirstName	LastName	Salary
1	Anna	Nowak	250000.00
2	Piotr	Zieliński	150000.00
3	Agnieszka	Kowalczyk	140000.00

**6. Korzystając z CTE, przygotuj zestawienie: EmployeeID, FirstName, LastName, DepartmentName, Salary — najpierw CTE pobierające średnią pensję w każdym dziale, a potem dołącz tabelę pracowników i wyświetl tylko tych, którzy zarabiają powyżej średniej w swoim dziale.**

```
WITH DeptAvgSalary AS (
    SELECT DepartmentID, AVG(Salary) AS AvgSalary
    FROM Employees
    GROUP BY DepartmentID
)
```

```
SELECT e.EmployeeID, e.FirstName, e.LastName, d.DepartmentName, e.Salary
FROM Employees e
```

```

JOIN DeptAvgSalary das ON e.DepartmentID = das.DepartmentID
JOIN Departments d ON e.DepartmentID = d.DepartmentID
WHERE e.Salary > das.AvgSalary
ORDER BY d.DepartmentName, e.Salary DESC;

```

	EmployeeID	FirstName	LastName	DepartmentName	Salary
1	1	Anna	Nowak	HR	250000.00
2	2	Piotr	Zieliński	IT	150000.00
3	5	Agnieszka	Kowalczyk	Sales	140000.00

**7. (Bonus) opcjonalnie — jeżeli zaimplementujesz kolumnę ManagerID, napisz zapytanie z rekurencyjnym CTE, które dla zadanego ManagerID = 1 wypisze całą strukturę podległych pracowników (hierarchię).**

```

WITH EmployeeHierarchy (EmployeeID, FirstName, LastName, ManagerID, Level)
AS (
    SELECT EmployeeID, FirstName, LastName, ManagerID, 1 AS Level
    FROM Employees
    WHERE ManagerID = 1

    UNION ALL

    SELECT e.EmployeeID, e.FirstName, e.LastName, e.ManagerID, eh.Level + 1
    FROM Employees e
    INNER JOIN EmployeeHierarchy eh ON e.ManagerID = eh.EmployeeID
)
SELECT EmployeeID, FirstName, LastName, ManagerID, Level
FROM EmployeeHierarchy
ORDER BY Level, ManagerID, EmployeeID;

```

	EmployeeID	FirstName	LastName	ManagerID	Level
1	2	Piotr	Zieliński	1	1
2	5	Agnieszka	Kowalczyk	1	1
3	7	Katarzyna	Dąbrowska	1	1
4	3	Marta	Wiśniewska	2	2
5	4	Tomasz	Wójcik	2	2
6	10	Dorota	Jankowska	2	2
7	6	Michał	Lewandowski	5	2
8	9	Łukasz	Kozłowski	5	2
9	8	Paweł	Kamiński	7	2