

Bazy Danych - Laboratorium 5

Student:

Bartłomiej Krawczyk

Numer Albumu:

310774

```
SELECT * FROM employees CROSS JOIN departments;
```

-- Pokaż wszystkie kombinacje pracowników (employees) oraz uzyskanych ocen z oceny rocznej (grades). Pokaż identyfikator pracownika oraz ocenę liczbową i jej opis.

```
SELECT employee_id, grade, description FROM employees, grades;
```

-- Zmodyfikuj poprzednie zapytanie tak aby pokazać tylko pracowników z departamentów 101, 102, 103 lub bez departamentu.

```
SELECT employee_id, grade, description
FROM employees, grades
WHERE department_id IN (101, 102, 103) OR department_id IS NULL;
```

```
SELECT * FROM employees e
INNER JOIN departments d
ON (e.department_id = d.department_id);
```

```
SELECT e.name, d.name FROM employees e
JOIN departments d
USING (department_id);
```

```
SELECT e.name, d.name FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

```
SELECT *
FROM countries
NATURAL JOIN addresses;
```

-- Znajdź pracowników, których zarobki nie są zgodne z "widełkami" na jego

stanowisku. Zwróć imię, nazwisko, wynagrodzenie oraz nazwę stanowiska.

```
SELECT e.name, e.surname, e.salary, p.name Position FROM employees e
JOIN positions p USING (position_id)
WHERE e.salary > p.max_salary OR e.salary < p.min_salary;
```

-- Zmodyfikuj poprzednie zapytanie tak, aby dodatkowo wyświetlić informacje o nazwie zakładu pracownika.

```
SELECT e.name, e.surname, e.salary, p.name Position, d.name Department FROM
employees e
JOIN positions p USING (position_id)
JOIN departments d USING (department_id)
WHERE e.salary > p.max_salary OR e.salary < p.min_salary;
```

-- Wyświetl nazwę zakładu wraz z imieniem i nazwiskiem jego kierowników. Pokaż tylko zakłady, które mają budżet pomiędzy 5000000 i 10000000.

```
SELECT d.name, e.name, e.surname FROM departments d
JOIN employees e ON (e.employee_id = d.manager_id)
WHERE d.year_budget BETWEEN 5000000 AND 10000000;
```

```
SELECT * FROM departments
NATURAL JOIN employees
WHERE year_budget BETWEEN 5000000 AND 10000000;
```

-- Nie da się ponieważ jest to równoważne

```
SELECT * FROM departments
JOIN employees USING (name, manager_id)
WHERE year_budget BETWEEN 5000000 AND 10000000;
```

-- Znajdź zakłady (podaj ich nazwę), które mają swoje siedziby w Polsce.

```
SELECT d.name FROM departments d
JOIN addresses a USING (address_id)
JOIN countries c USING (country_id)
WHERE c.name = 'Polska';
```

-- Zmodyfikuj zapytanie 3 tak, aby uwzględniać w wynikach tylko zakłady, które mają siedziby w Polsce.

```
SELECT d.name, e.name, e.surname FROM departments d
JOIN employees e ON (e.employee_id = d.manager_id)
JOIN addresses a USING (address_id)
JOIN countries c USING (country_id)
WHERE
    d.year_budget BETWEEN 5000000 AND 10000000
    AND
    c.name = 'Polska';
```

-- Pokaż oceny (grades) pracowników którzy nie posiadają kierownika. W wynikach

pokaż imię , nazwisko pracownika, ocene liczbowa i jej opis.

```
SELECT e.name, e.surname, g.grade, g.description FROM employees e
JOIN emp_grades USING (employee_id)
JOIN grades g USING (grade_id)
WHERE e.manager_id IS NULL;
```

-- Pokaż nazwę kraju i nazwę regionu do którego został przypisany.

```
SELECT c.name Country, r.name Region FROM countries c
JOIN reg_countries USING (country_id)
JOIN regions r USING (region_id);
```

-- Wyświetl listę zawierającą nazwisko pracownika, stanowisko, na którym pracuje, aktualne zarobki oraz widełki płacowe dla tego stanowiska.

-- Sterując rodzajem złączenia, zagwarantuj, że w wynikach znajdą się wszyscy pracownicy.

```
SELECT e.surname, p.name, e.salary, p.min_salary, p.max_salary
FROM employees e
LEFT JOIN positions p USING (position_id);
```

-- Wyświetl średnią pensję oraz liczbę osób zatrudnionych dla stanowisk.

-- Sterując rodzajem złączenia zagwarantuj, że znajdą się tam również stanowiska, na których nikt nie jest zatrudniony.

```
SELECT AVG(e.salary), COUNT(employee_id), p.name FROM employees e
RIGHT JOIN positions p USING (position_id)
GROUP BY position_id, p.name;
```

-- Pokaż liczbę pracowników zatrudnionych kiedykolwiek w każdym projekcie. Zadbaj by w wynikach pojawił się każdy projekt.

```
SELECT p.name, COUNT(employee_id) FROM employees
JOIN EMP_PROJECTS USING (employee_id)
RIGHT JOIN PROJECTS p USING (project_id)
GROUP BY project_id, p.name;
```

-- Pokaż średnią ocenę pracowników per departament. W wynikach zamieść nazwę departamentu i średnią ocenę.

```
SELECT d.name, AVG(g.grade) FROM grades g
JOIN emp_grades USING (grade_id)
JOIN employees e USING (employee_id)
RIGHT JOIN departments d USING (department_id)
GROUP BY department_id, d.name;
```

-- Dla każdego imienia pracownika z zakładów Administracja lub Marketing zwróć liczbę pracowników, którzy mają takie samo imię i podaj ich średnie zarobki.

```
SELECT e.name, COUNT(e.name), AVG(e.salary) FROM employees e
JOIN departments d USING (department_id)
WHERE d.name IN ('Administracja', 'Marketing')
GROUP BY e.name
HAVING COUNT(e.name) > 1;
```

-- Zwróć imiona i nazwiska pracowników, którzy przeszli więcej niż 2 zmiany stanowiska. Wyniki posortuj malejąco wg liczby zmian.

```
SELECT e.name, e.surname, COUNT(*) changes FROM employees e
JOIN positions_history USING (employee_id)
GROUP BY (employee_id, e.name, e.surname)
HAVING COUNT(*) > 2
ORDER BY changes DESC;
```

-- Zwróć id, nazwisko kierowników oraz liczbę podległych pracowników. Wyniki posortuj malejąco wg liczby podległych pracowników.

```
SELECT m.employee_id, m.name, m.surname, COUNT(*) FROM employees e
JOIN employees m ON (m.employee_id = e.manager_id)
GROUP BY m.employee_id, m.name, m.surname
ORDER BY COUNT(*) DESC;
```

-- Napisz zapytanie zwracające liczbę zakładów w krajach. W wynikach podaj nazwę kraju oraz jego ludność.

```
SELECT c.name, c.population, COUNT(*) Departments FROM departments d
JOIN addresses a USING (address_id)
RIGHT JOIN countries c USING (country_id)
GROUP BY country_id, c.name, c.population;
```

-- Napisz zapytanie zwracające liczbę zakładów w regionach. W wynikach podaj nazwę regionu. Wynik posortuj malejąco względem liczby zakładów.

```
SELECT r.name, COUNT(*) Departments FROM departments d
JOIN addresses a USING (address_id)
RIGHT JOIN countries c USING (country_id)
RIGHT JOIN reg_countries USING (country_id)
RIGHT JOIN regions r USING (region_id)
GROUP BY region_id, r.name
ORDER BY COUNT(*) DESC;
```

-- Napisz zapytanie znajdujące liczbę zmian stanowisk pracownika Jan Kowalski.

```
SELECT e.name, e.surname, COUNT(*) FROM positions_history p
RIGHT JOIN employees e USING (employee_id)
WHERE e.name = 'Jan' AND e.surname = 'Kowalski'
GROUP BY employee_id, e.name, e.surname;
```

-- Napisz zapytanie znajdujące średnią pensję dla każdego ze stanowisk. Wynik powinien zawierać nazwę stanowiska i zaokrągloną średnią pensję.

```
SELECT p.name, ROUND(AVG(e.salary)) FROM positions p
JOIN employees e USING (position_id)
GROUP BY position_id, p.name;
```

-- Pobierz wszystkich pracowników zakładu Kadry lub Finanse wraz z informacją w jakim zakładzie pracują.

```
SELECT e.name, e.surname, d.name Department FROM employees e
JOIN departments d USING (department_id)
WHERE d.name IN ('Kadry', 'Finanse');
```

-- Znajdź pracowników, których zarobki nie są zgodne z “widełkami” na jego stanowisku. Zwróć imię, nazwisko, wynagrodzenie oraz nazwę stanowiska. Zrealizuj za pomocą złączenia nierównościowego.

```
SELECT e.name, e.surname, e.salary, p.name FROM employees e
LEFT JOIN positions p USING (position_id)
WHERE e.salary > p.max_salary OR e.salary < p.min_salary;
```

-- Pokaż nazwy regionów w których nie ma żadnego kraju.

```
SELECT r.name, COUNT(*) FROM regions r
LEFT JOIN reg_countries USING (region_id)
LEFT JOIN countries c USING (country_id)
GROUP BY region_id, r.name
HAVING COUNT(*) = 0;
```

-- Wykonaj złączenie naturalne między tabelami countries a regions. Jaki wynik otrzymujemy i dlaczego?

```
SELECT * FROM countries
NATURAL JOIN regions;
```

-- Jako wynik otrzymujemy pustą tabelę
-- takie wywołanie jest równoważne:

```
SELECT * FROM countries
JOIN regions USING (name);
```

-- a nie ma kraju o nazwie takiej samej jak region

-- Jaki otrzymamy wynik jeśli zrobimy NATURAL JOIN na tabelach bez wspólnej kolumny? Sprawdź i zastanów się nad przyczyną

```
SELECT * FROM regions
NATURAL JOIN positions_history;
```

-- dostajemy wszystkie możliwe kombinacje - CROSS JOIN
-- ponieważ jest to równoważne wywołaniu JOIN bez wykorzystania żadnych kluczy:

```
SELECT * FROM regions
JOIN positions_history ON 1=1;
```

```
SELECT * FROM regions  
CROSS JOIN positions_history;
```