

# Bazy Danych - Laboratorium 4

---

Student:

Bartłomiej Krawczyk

Numer Albumu:

310774

-- Przygotuj zapytanie, które wyświetli informację ilu pracowników ma aktualnie dany status\_id (Status\_ID odwołujący się do tabeli EMP\_STATUS).

```
SELECT status_id, COUNT(*) FROM employees
GROUP BY status_id;
```

-- Zmodyfikuj poprzednie zapytanie, żeby pokazać jedynie liczbę kobiet będących w danym statusie.

```
SELECT status_id, COUNT(*) FROM employees
WHERE gender LIKE 'K'
GROUP BY status_id;
```

-- Wyświetl minimalne, maksymalne zarobki, a także średnią, medianę i odchylenie standardowe zarobków pracowników na każdym ze stanowisk (wykorzystaj tylko tabelę Employees).

```
SELECT position_id, MIN(salary), MAX(salary), AVG(salary)
FROM employees
GROUP BY position_id;
```

-- Napisz zapytanie, które dla określonego języka zwróci: liczbę krajów które używają tego języka, średnią populację.

```
SELECT language, COUNT(*), AVG(population)
FROM countries
GROUP BY language;
```

-- Dla każdej z płci oblicz średnią pensję, średni wiek oraz średnią długość zatrudnienia. Wyniki posortuj względem średniej pensji malejąco.

```
SELECT gender, AVG(salary), AVG(FLOOR(MONTHS_BETWEEN(SYSDATE, birth_date) / 12))
AGE, AVG(salary), AVG(MONTHS_BETWEEN(SYSDATE, date_employed) / 12)
YEARS_OF_EMPLOYMENT
FROM employees
```

```
GROUP BY gender
ORDER BY AVG(salary) DESC;

-- Oblicz liczbę założonych departamentów w każdym roku.

SELECT EXTRACT(YEAR FROM established) YEAR, COUNT(*)
FROM departments
GROUP BY EXTRACT(YEAR FROM established);

-- Oblicz liczbę pracowników zatrudnionych każdego miesiąca(sty, lu, ma..)

SELECT EXTRACT(MONTH FROM date_employed), COUNT(*)
FROM employees
GROUP BY EXTRACT(MONTH FROM date_employed);

SELECT TO_CHAR(date_employed, 'MONTH'), COUNT(*)
FROM employees
GROUP BY TO_CHAR(date_employed, 'MONTH');

-----

-- Wyświetl informacje o liczbie krajów mających dany język jako urzędowy. Pokaż
języki które są wykorzystane przez przynajmniej 2 kraje.

SELECT language, COUNT(*)
FROM countries
GROUP BY language
HAVING COUNT(*) >= 2;

-- Wyświetl średnie zarobki dla każdego ze stanowisk, o ile średnie te są większe
od 2000.

SELECT position_id, AVG(salary)
FROM employees
GROUP BY position_id
HAVING AVG(salary) > 2000;

-- Wyświetl średnie zarobki dla każdego ze stanowisk, o ile średnie te są większe
od 2000 i liczba pracowników na danym stanowisku jest większa niż 1.

SELECT position_id, AVG(salary)
FROM employees
GROUP BY position_id
HAVING AVG(salary) > 2000 AND COUNT(*) > 1;

-- Wyświetl średnie zarobki dla wszystkich pracowników pogrupowane ze względu na
kolumny Department_ID, Status_ID, o ile ich Status_ID = 301 lub 304.
-- Porównaj rezultaty zapytania jeśli warunek ograniczający Status_ID jest
umieszczony:
-- w klauzuli WHERE,

SELECT department_id, status_id, AVG(salary)
FROM employees
WHERE status_id IN (301, 304)
```

```
GROUP BY department_id, status_id;

-- w klauzuli HAVING
SELECT department_id, status_id, AVG(salary)
FROM employees
GROUP BY department_id, status_id
HAVING status_id IN (301, 304);

-- Porównanie
SELECT * FROM
(SELECT department_id, status_id, AVG(salary) A_salary FROM employees WHERE
status_id IN (301, 304) GROUP BY department_id, status_id) A
FULL OUTER JOIN
(SELECT department_id, status_id, AVG(salary) B_salary FROM employees GROUP BY
department_id, status_id HAVING status_id IN (301, 304)) B
ON A.department_id = B.department_id AND A.status_id = B.status_id
WHERE A_salary != B_salary;

-- Otrzymujemy dokładnie takie same rezultaty, jednak bardziej optymalna jest
wersja a) ponieważ WHERE jest wykonywane przed GROUP BY, a HAVING dopiero po GROUP
BY
-- ==> TO oznacza, że do GROUP BY przekazujemy większą listę w zapytaniu b) niż w
a)

-----

-- Napisz polecenie które zwróci nazwę regionu i jego nazwę skróconą oraz nazwę
kraju oraz jego nazwę skróconą. Rozróżnij regiony od krajów dodając kolumnę rodzaj
przyjmującą wartości "R" dla regionów i "K" dla krajów.

SELECT name, shortname, 'R' kind FROM regions
UNION SELECT name, code, 'K' FROM countries;

-- Napisz polecenie które zwróci imię nazwisko i wiek pracowników oraz imię,
nazwisko i wiek dzieci pracowników. Rozróżnij pracowników od dzieci dodając
kolumnę rodzaj przyjmującą wartości "P" dla pracowników i "D" dla dzieci.

SELECT name, surname, FLOOR(MONTHS_BETWEEN(SYSDATE, birth_date) / 12) AGE, 'P'
kind FROM employees
UNION SELECT name, surname, FLOOR(MONTHS_BETWEEN(SYSDATE, birth_date) / 12), 'D'
FROM dependents;

-- Korzystając z operatora UNION napisz zapytanie, które zwróci id, imię i
nazwisko wszystkich pracowników pracujących w zakładzie o ID = 101 lub na
stanowisku o ID = 103. (Jak można to inaczej zapisać? Jak myślisz która wersja
jest wydajniejsza?)

SELECT employee_id, name, surname FROM employees
WHERE department_id = 101
UNION
SELECT employee_id, name, surname FROM employees
WHERE position_id = 103;

-- lub
```

```
SELECT employee_id, name, surname
FROM employees
WHERE department_id = 101 OR position_id = 103;
```

-- Jestem pewny, że druga opcja jest wydajniejsza, ponieważ w pierwszej wersji wywołujemy dwa SELECT-y i dopiero na nich wykonujemy operację UNION, a w drugiej wersji wykonujemy jeden SELECT i sprawdzamy dwa warunki na raz

-- Korzystając z operatora INTERSECT pokaż nazwy wszystkich stanowisk, które rozpoczynają się od liter P, K lub A, a minimalne zarobki (według tabeli POSITIONS) są dla nich większe lub równe 1500.

```
SELECT name FROM positions
WHERE SUBSTR(name, 1, 1) IN ('P', 'K', 'A')
INTERSECT
SELECT name FROM positions
WHERE min_salary >= 1500;
```

-- Z zastosowaniem operatora MINUS wyświetl średnie zarobki (dla tabeli Employees) dla wszystkich stanowisk z wyłączeniem stanowiska o ID = 102. Posortuj rezultat malejąco według średnich zarobków.

```
SELECT AVG(salary) sal FROM employees
GROUP BY position_id
MINUS
SELECT AVG(salary) FROM employees
WHERE position_id = 102
GROUP BY position_id
ORDER BY sal DESC NULLS LAST;
```

-----

-- Wyznacz średnie zarobki pracowników ze względu na zakłady, o ile są to pracownicy zatrudnieni przed 01.01.2020. Następnie dodatkowo ogranicz powyższe zapytanie do tych zakładów, które zatrudniają więcej niż 2 takie osoby.

```
SELECT department_id, AVG(salary) FROM employees
WHERE date_employed < '10/01/01'
GROUP BY department_id
HAVING COUNT(*) > 2;
```

-- Wyznacz średnie zarobki pracowników ze względu na zakłady, o ile są to pracownicy zatrudnieni przed 01.01.2010. Dodatkowo ogranicz powyższe zapytanie do tych zakładów, które zatrudniają więcej niż 2 osoby (w ogóle, a nie tylko takie, które zostały zatrudnione przed 01.01.2010)

```
SELECT department_id, AVG(salary) FROM employees
WHERE date_employed < '10/01/01'
GROUP BY department_id
MINUS
SELECT department_id, AVG(salary) FROM employees
GROUP BY department_id
HAVING COUNT(*) < 3;
```

-- Napisz zapytanie które dla każdego departamentu wyświetli średnią pensję w zależności od płci.

```
SELECT department_id, AVG(salary), gender FROM employees
GROUP BY department_id, gender;
```

-- Napisz zapytanie które pogrupuje liczby krajów ze względu na pierwszą literę nazwy języka używanego w danym kraju.

```
SELECT SUBSTR(name, 1, 1), COUNT(*) FROM countries
GROUP BY SUBSTR(name, 1, 1);
```

-- Polecenie SELECT name, surname, COUNT(\*) FROM employees GROUP BY name HAVING COUNT(\*) >=2; jest niepoprawne. Dlaczego?

```
SELECT name, surname, COUNT(*) FROM employees
GROUP BY name HAVING COUNT(*) >=2;
```

-- Ponieważ w wywołaniach GROUP BY w SELECT możemy jedynie korzystać z funkcji agregujących lub kolumn które zostały użyte do grupowania

-- Dla każdego departamentu zwróć informację o maksymalnej pensji pracownika z tego departamentu.

```
SELECT department_id, MAX(salary) FROM employees
GROUP BY department_id;
```

-- Ile walut jest oficjalną walutą wykorzystywaną w więcej niż 1 kraju?

```
SELECT COUNT(*)
FROM (SELECT currency FROM countries
GROUP BY currency
HAVING COUNT(*) > 1);
```

-- Ile jest średnio zmian na stanowiskach (skorzystaj z positions\_history)?

```
SELECT AVG(count) FROM
(SELECT position_id, COUNT(*) count FROM positions_history GROUP BY position_id);
```

-- Przy grupowaniu danych wykorzystując jedną kolumnę, ile powstanie grup danych?

-- Powstanie tyle grup danych co unikalnych wartości w danej kolumnie np.

```
SELECT DISTINCT name FROM employees;
```

```
SELECT name FROM employees
GROUP BY name;
```