



**Wydział Elektroniki
i Technik Informatycznych**

POLITECHNIKA WARSZAWSKA

Bazy Danych 1

edycja 21Z

Laboratorium 4

Wprowadzenie i przebieg laboratorium

- Grupowanie danych (klauzule GROUP BY oraz HAVING, funkcje agregujące)
- Łączenie rezultatów zapytań za pomocą operatorów UNION, INTERSECT oraz MINUS

Grupowanie danych za pomocą klauzuli GROUP BY

- Klauzula GROUP BY pozwala na pogrupowanie danych ze względu na wartości jednej lub więcej kolumn. Załóżmy, że dany jest zbiór pracowników jak poniżej i chcemy otrzymać średnią zarobków w każdym z zakładów.

Tabela Employees

Employee_ID	Salary	Department_ID
1	3000	101
2	1400	102
3	2000	102
4	1300	102
5	500	103
6	2400	101

Zapytanie grupujące

```
SELECT AVG(Salary), Department_ID  
FROM Employees  
GROUP BY Department_ID;
```

Rezultat zapytania

AVG(Salary)	Department_ID
2700	101
1566.67	102
500	103

Group BY a wiele kolumn grupujących

→ Możliwe jest użycie wielu kolumn grupujących np.: dany jest zbiór pracowników jak poniżej i chcemy otrzymać średnią zarobków w ze względu na stanowiska w każdym z zakładów.

Tabela Employees

Employee_ID	Salary	Department_ID	Position_ID
1	3000	101	1
2	1400	102	1
3	2000	102	1
4	1300	102	2
5	500	101	1
6	2400	101	3

Zapytanie grupujące

```
SELECT AVG(Salary),  
       Department_ID, Position_ID  
FROM Employees  
GROUP BY Department_ID,  
         Position_ID;
```

Rezultat zapytania

AVG(Salary)	Department_ID	Position_ID
1750	101	1
2400	101	3
1566.67	102	1
1300	102	2

GROUP BY - uwagi i funkcje grupujące

1. Klauzula GROUP BY wykonywana jest zawsze po klauzuli WHERE.
2. Po zastosowaniu grupowania za pomocą GROUP BY można odwoływać się tylko do **kolumn, które zostały zastosowane do grupowania** lub używać **funkcji grupujących**.

Niektóre funkcje grupujące i ich opisy

Nazwa funkcji	Opis
AVG(X)	Zwraca średnią dla kolumny X w każdej z grup (X niekoniecznie musi być kolumną grupującą)
COUNT(X)	Zwraca liczbę wierszy kolumny X w każdej z grup
MAX(X), MIN(X)	Zwraca maksymalną i minimalną wartość w kolumnie X dla każdej z grup
MEDIAN(X)	Zwraca medianę wartości w kolumnie X dla każdej z grup
STDDEV(X)	Odchylenie standardowe wartości w kolumnie X dla każdej z grup
SUM(X)	Suma wartości w kolumnie X dla każdej z grup

GROUP BY - grupowanie według wyrażenia

1. Klauzula GROUP BY niekoniecznie musi zawierać tylko nazwy kolumn - mogą tam również znaleźć się wyrażenia (np. funkcje wierszowe zastosowane dla wartości kolumny).
2. Przykład: *wyznacz średni roczny budżet departamentów ze względu na rok ich założenia.*

```
SELECT avg(year_budget)
FROM departments
GROUP BY extract(year from established);
```

Ćwiczenia - grupowanie

1. Przygotuj zapytanie, które wyświetli informację ilu pracowników ma aktualnie dany status_id (Status_ID odwołujący się do tabeli EMP_STATUS).
2. Zmodyfikuj poprzednie zapytanie, żeby pokazać jedynie liczbę kobiet będących w danym statusie.
3. Wyświetl minimalne, maksymalne zarobki, a także średnią, medianę i odchylenie standardowe zarobków pracowników na każdym ze stanowisk (wykorzystaj tylko tabelę Employees).
4. Napisz zapytanie, które dla określonego języka zwróci: liczbę krajów które używają tego języka, średnią populację.
5. Dla każdej z płci oblicz średnią pensję, średni wiek oraz średnią długość zatrudnienia. Wyniki posortuj względem średniej pensji malejąco.
6. Oblicz liczbę założonych departamentów w każdym roku.
7. Oblicz liczbę pracowników zatrudnionych każdego miesiąca(sty, lu, ma..)

HAVING - ograniczenia na zwracane grupy

1. Klauzula HAVING zawsze występuje w połączeniu z GROUP BY i może zawierać warunki selekcji grup. Uwaga! Klauzula HAVING może zawierać jedynie warunki logiczne z **kolumnami, które zostały użyte do grupowania lub funkcje agregujące**.
2. *Przykład:* dany jest zbiór pracowników jak poniżej i chcemy otrzymać średnią zarobków w ze względu na stanowiska w każdym z zakładów, o ile dotyczy to stanowisk o ID = 1 i 3.

Employee_ID	Salary	Department_ID	Position_ID
1	3000	101	1
2	1400	102	1
3	2000	102	1
4	1300	102	2
5	500	101	1
6	2400	101	3

```
SELECT AVG(Salary),  
       Department_ID, Position_ID  
FROM Employees  
GROUP BY Department_ID,  
         Position_ID  
HAVING Position_ID IN (1, 3);
```

AVG(Salary)	Department_ID	Position_ID
1750	101	1
2400	101	3
1566.67	102	1

HAVING - zastosowanie funkcji grupującej

1. W ramach klauzuli HAVING można umieścić funkcje grupujące (ale nie funkcje wierszowe).
2. *Przykład:* dany jest zbiór pracowników jak poniżej i chcemy otrzymać średnią zarobków w ze względu na stanowiska w każdym z zakładów, o ile dotyczy to stanowisk o ID = 1 i 3 i o ile ta średnia jest większa od 1700.

Employee_ID	Salary	Department_ID	Position_ID
1	3000	101	1
2	1400	102	1
3	2000	102	1
4	1300	102	2
5	500	101	1
6	2400	101	3

```
SELECT AVG(Salary),  
       Department_ID, Position_ID  
FROM Employees  
GROUP BY Department_ID,  
         Position_ID  
HAVING Position_ID IN (1, 3)  
       AND AVG(Salary) > 1700;
```

AVG(Salary)	Department_ID	Position_ID
1750	101	1
2400	101	3

Ćwiczenia - HAVING

1. Wyświetl informacje o liczbie krajów mających dany język jako urzędowy. Pokaż języki które są wykorzystane przez przynajmniej 2 kraje.
2. Wyświetl średnie zarobki dla każdego ze stanowisk, o ile średnie te są większe od 2000.
3. 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.
4. 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:
 - a. w klauzuli WHERE,
 - b. w klauzuli HAVING.

Łączenie rezultatów zapytań za pomocą operatorów UNION, INTERSECT oraz MINUS

→ Operatory UNION [ALL], INTERSECT i MINUS pozwalają na wykonywanie operacji sumy, przecięcia oraz różnicy zbiorów rezultatów zapytań SELECT. Załóżmy, że otrzymaliśmy dwa rezultaty zapytań jak poniżej:

Rezultat dla SELECT 1		
Name	Surname	Position_ID
Andrzej Piotr	Zalewski Nowak	1 2
Krzysztof Barbara	Wiech Abacka	3 4

Rezultat dla SELECT 2		
Name	Surname	Position_ID
Krzysztof Barbara	Wiech Abacka	3 4
Jacek Agata	Nowak Nowak	5 5

Wiersze w czerwonych ramkach powtarzają się w obu rezultatach

Łączenie rezultatów zapytań za pomocą operatorów UNION, INTERSECT oraz MINUS (cd.)

- Ponieważ sekwencja **typów** danych dla kolumn wynikowych zapytań SELECT 1 oraz SELECT 2 jest taka sama, to możliwe jest połączenie rezultatów zapytań za pomocą operatorów UNION [ALL], INTERSECT oraz MINUS.
- UNION usuwa duplikaty wierszy w wynikach zapytania, UNION ALL pozostawia duplikaty.

Rezultat dla SELECT 1 UNION SELECT 2		
Name	Surname	Position_ID
Andrzej	Zalewski	1
Piotr	Nowak	2
Krzysztof	Wiech	3
Barbara	Abacka	4
Jacek	Nowak	5
Agata	Nowak	5

Rezultat dla SELECT 1 UNION ALL SELECT 2		
Name	Surname	Position_ID
Andrzej	Zalewski	1
Piotr	Nowak	2
Krzysztof	Wiech	3
Barbara	Abacka	4
Krzysztof	Wiech	3
Barbara	Abacka	4
Jacek	Nowak	5
Agata	Nowak	5

Rezultat dla SELECT 1 INTERSECT SELECT 2		
Name	Surname	Position_ID
Krzysztof	Wiech	3
Barbara	Abacka	4

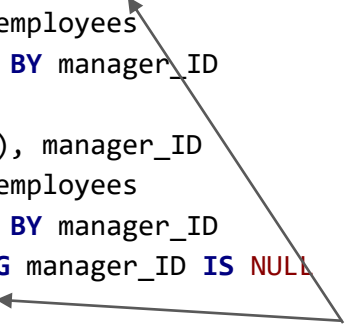
Rezultat dla SELECT 1 MINUS SELECT 2		
Name	Surname	Position_ID
Andrzej	Zalewski	1
Piotr	Nowak	2

Operatory UNION (ALL), INTERSECT, MINUS - przykład sortowania z ORDER BY

→ Przykład: Posortuj rosnąco średnie zarobki pracowników grupując ich ze względu na ich managerów (usuń grupę pracowników, którzy nie mają managera).

Rozwiązanie 1: Zastosowanie aliasu

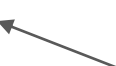
```
SELECT AVG(salary) avg_sal, manager_ID
      FROM employees
      GROUP BY manager_ID
MINUS
SELECT AVG(salary), manager_ID
      FROM employees
      GROUP BY manager_ID
      HAVING manager_ID IS NULL
ORDER BY avg_sal;
```



Używamy aliasu w przypadku zastosowania funkcji lub wyrażenia, według którego rezultat ma być posortowany.

Rozwiązanie 2: Zastosowanie nr kolumny

```
SELECT AVG(salary), manager_ID
      FROM employees
      GROUP BY manager_ID
MINUS
SELECT AVG(salary), manager_ID
      FROM employees
      GROUP BY manager_ID
      HAVING manager_ID IS NULL
ORDER BY 1;
```



Alternatywnie możemy użyć numeru kolumny z rezultatu zapytania.

Ćwiczenia - operatory UNION (ALL), INTERSECT, MINUS

1. 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.
2. 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.
3. 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?)*
4. 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.
5. 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.



Praca domowa

1. Zapoznaj się z materiałami Oracle Academy dotyczącymi grupowania oraz operacji na zbiorach.
2. 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.
3. 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)
4. Napisz zapytanie które dla każdego departamentu wyświetli średnią pensję w zależności od płci.
5. Napisz zapytanie które wyznaczy liczbę krajów w danym regionie które mają język urzędowy angielski



Praca domowa

6. Polecenie `SELECT name, surname, COUNT(*) FROM employees GROUP BY name HAVING COUNT(*) >=2;` jest niepoprawne. Dlaczego?
7. Dla każdego departamentu zwróć informację o maksymalnej pensji pracownika z tego departamentu.
8. Ile walut jest oficjalną walutą wykorzystywaną przez więcej niż 1 kraj?
9. Ile jest średnio zmian na stanowiskach (skorzystaj z `positions_history`)?
10. Przy grupowaniu danych przez jedną kolumnę ile powstanie grup danych?