

## PLAN ZESTAWU

1. Charakterystyka funkcji agregujących
2. Przegląd dostępnych funkcji agregujących, sposób użycia
3. Konstruowanie zapytań z jedną grupą
4. Konstruowanie zapytań z wieloma grupami
5. Podział grup na podgrupy
6. Filtrowanie grup
7. Najczęstsze błędy przy konstrukcji zapytań z funkcjami agregującymi
8. Przegląd zaawansowanych konstrukcji z funkcjami agregującymi
9. Zadania

### 1. Charakterystyka funkcji agregujących

- Działają na zbiorach rekordów, nazywanych grupami.
- Rekordy należą do tej samej grupy jeśli posiadają tę samą wartość wyrażenia grupującego.
- Funkcja agregująca dla każdej grupy wylicza pojedynczą wartość na podstawie wyrażenia, będącego jej parametrem.

#### **Przykładowy problem:**

*znajdź średnią płacę pracowników dla każdej grupy etatowej.*

### 2. Przegląd dostępnych funkcji agregujących, sposób użycia

- Funkcje:
  - AVG – średnia,
  - MIN – minimum,
  - COUNT – liczba wystąpień,
  - SUM – suma,
  - MAX – maksimum,
- Składnia: nazwa\_funkcji (all | distinct wyrażenie)
- Szczególny przypadek – funkcja COUNT:
  - COUNT(\*) – liczba rekordów,
  - COUNT(all | distinct wyrażenie) – liczba niepustych wartości wyrażenia.

### 3. Konstruowanie zapytań z jedną grupą

- Wszystkie rekordy, przetwarzane przez zapytanie, tworzą jedną grupę.
- Wynikiem zapytania jest jeden rekord.

#### **Przykładowy problem:**

*znajdź minimalną i maksymalną wartość płacy podstawowej pracowników zespołu o identyfikatorze 20.*

### 4. Konstruowanie zapytań z wieloma grupami

## Zestaw 3

Klauzula GROUP BY wyrażenie – definiuje wyrażenie grupujące. Wynikiem zapytania jest jeden rekord dla każdej grupy.

```
SELECT atr1, ... , atrn  
FROM nazwa_tabeli  
GROUP BY atr_k
```

### **Przykładowy problem:**

znajdź średnią wartość płacy podstawowej wśród pracowników każdego zespołu, posortuj wg identyfikatora zespołu.

## 5. Podział grup na podgrupy

### **Przykładowy problem:**

dla każdego zespołu znajdź liczbę pracowników pracujących na poszczególnych etatach.

## 6. Filtrowanie grup

- Klauzula HAVING warunek\_logiczny – umożliwia wybór grup, spełniających warunek logiczny.
- Warunek logiczny może być skonstruowany jedynie z funkcji agregujących i/lub wyrażień grupujących.

```
SELECT atr1, ... , atrn  
FROM nazwa_tabeli  
GROUP BY atr_k  
HAVING warunek_logiczny
```

### **Przykładowy problem:**

podaj identyfikatory i średnie płace podstawowe w zespołach zatrudniających nie mniej niż trzech pracowników.

## 7. Najczęstsze błędy przy konstrukcji zapytań z funkcjami agregującymi

- Umieszczenie w klauzuli SELECT zapytania z jedną grupą wyrażenia nie będącego funkcją agregującą.  
**SELECT etat, SUM(placa\_pod) FROM pracownicy WHERE etat='PROFESOR';**
- Umieszczenie w klauzuli SELECT zapytania z wieloma grupami wyrażenia nie będącego wyrażeniem grupującym lub funkcją agregującą.  
**SELECT id\_zesp, nazwisko, SUM(placa\_pod) FROM pracownicy GROUP BY id\_zesp;**
- Umieszczanie funkcji agregującej w klauzuli WHERE.  
**SELECT id\_zesp FROM pracownicy WHERE COUNT(\*)>3 GROUP BY id\_zesp;**

## Zestaw 3

- Umieszczenie w warunku klauzuli **HAVING** wyrażenia nie będącego funkcją agregującą lub wyrażeniem grupującym .

```
SELECT id_zesp, COUNT(distinct placa_dod) FROM pracownicy GROUP BY id_zesp HAVING etat='PROFESOR';
```

- Porządkowanie wyników zapytania z wieloma grupami według wartości wyrażenia nie będącego wyrażeniem grupującym lub funkcją agregującą.

```
SELECT id_zesp, COUNT(distinct placa_dod) FROM pracownicy GROUP BY id_zesp ORDER BY nazwisko;
```

## 8. Przegląd zaawansowanych konstrukcji z funkcjami agregującymi

- Użycie funkcji agregującej jako parametru innej funkcji agregującej.

**Przykładowy problem:**

znajdź maksymalną sumę płac pracowników w poszczególnych zespołach.

- Zapytanie z jedną grupą i klauzulą **HAVING**;

**Przykładowy problem:**

podaj wartość średniej płacy pracowników, ale tylko wtedy, jeśli liczba pracowników jest większa od 12.

- Zapytanie z klauzulą **WHERE** i **HAVING**.

**Przykładowy problem:**

dla każdego zespołu, w którym średnia płaca przekracza 1000, podaj liczbę zatrudnionych pracowników, pominiętych pracowników na etacie PROFESOR, wynik uporządkuj ze względu na sumę płac podstawowych w zespole.

## 9. Zadania

- Wyświetl najniższą i najwyższą pensję oraz różnicę dzielącą najlepiej i najgorzej zarabiających pracowników.
- Wyświetl średnie pensje dla wszystkich etatów. Wyniki uporządkuj wg malejącej średniej pensji.
- Wyświetl liczbę zatrudnionych profesorów.
- Znajdź sumaryczne miesięczne płace dla każdego zespołu. Nie zapomnij o płacach dodatkowych!
- Wyświetl numery zespołów, które zatrudniają więcej niż dwóch pracowników. Pomiń pracowników bez przydziału do zespołów. Wyniki uporządkuj wg malejącej liczby pracowników.
- Wyświetl średnie pensje wypłacane w ramach poszczególnych etatów i liczbę pracowników zatrudnionych na danym etacie. Pomiń pracowników zatrudnionych po 1990 roku.
- Dla każdego pracownika wyświetl pensję najgorzej zarabiającego podwładnego. Wyniki uporządkuj wg malejącej pensji.
- Sprawdź, czy identyfikatory pracowników są unikalne.

## Podsumowanie

- Funkcje agregujące działają na zbiorach rekordów, nazywanych grupami.
- Funkcja agregująca wylicza pojedynczą wartość dla każdej grupy rekordów zapytania.
- Słowo kluczowe **GROUP BY** umożliwia podanie wyrażenia, którego wartości posłużą do podziału zbioru rekordów zapytania na grupy.
- Do eliminacji grup, nie spełniających określonych kryteriów, służy klauzula **HAVING**.