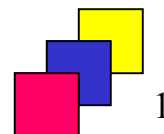


Język SQL. Rozdział 4.

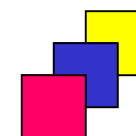
Funkcje grupowe

**Funkcje grupowe, podział relacji na grupy,
klauzule GROUP BY i HAVING.**



Funkcje grupowe (agregujące) (1)

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



Funkcje grupowe (agregujące) (2)

- **Problem:** znajdź sumaryczną płacę pracowników w każdej grupie etatowej.

wyrażenie grupujące:
etat

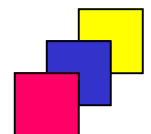
| ETAT | PLACA_POD |
|----------|-----------|
| ADIUNKT | 590 |
| ADIUNKT | 645,5 |
| ASYSTENT | 439,7 |
| ASYSTENT | 480 |
| ASYSTENT | 371 |
| ASYSTENT | 480 |
| PROFESOR | 1350 |
| PROFESOR | 830 |
| PROFESOR | 960 |
| PROFESOR | 1070 |

grupa dla etatu
= 'ADIUNKT'

grupa dla etatu
= 'ASYSTENT'

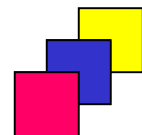
grupa dla etatu
= 'PROFESOR'

| ETAT | SUMA |
|----------|--------|
| ADIUNKT | 1235,5 |
| ASYSTENT | 1770,7 |
| PROFESOR | 4210 |



Rodzaje funkcji grupowych

- **Funkcje:**
 - **AVG** – średnia,
 - **COUNT** – liczba wystąpień,
 - **MAX** – maksimum,
 - **MIN** – minimum,
 - **SUM** – suma,
 - **VARIANCE** – wariancja,
 - **STDDEV** – odchylenie stand.
- **Składnia: nazwa_funkcji(all | distinct wyrażenie)**
- **Szczególny przypadek – funkcja COUNT:**
 - **COUNT(*)** – liczba rekordów,
 - **COUNT(all | distinct wyrażenie)** – liczba niepustych wartości wyrażenia.
- **Uwaga! Funkcje grupowe eliminują wartości puste.**

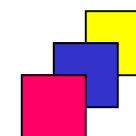
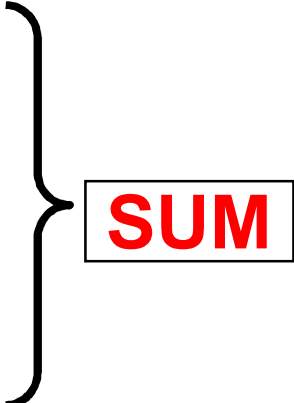


Zapytania z jedną grupą (1)

- Wszystkie rekordy, przetwarzane przez zapytanie, tworzą jedną grupę.
- Wynikiem zapytania jest co najwyżej jeden rekord.
- Znajdź sumaryczną płacę podstawową wszystkich pracowników.

```
SELECT SUM(placa_pod)  
FROM pracownicy;
```

| | PLACA_POD |
|-----|-----------|
| ... | 200 |
| ... | 340 |
| ... | null |
| ... | 456 |
| ... | 120 |



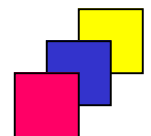
Zapytania z jedną grupą (2)

- Znajdź średnią płacę podstawową wśród pracowników zespołu 20.

```
SELECT AVG(placa_pod)  
FROM pracownicy  
WHERE id_zesp = 20;
```

- Znajdź maksymalną wartość dodatku oraz liczbę pracowników na etacie ASYSTENT.

```
SELECT COUNT(*), MAX(placa_dod)  
FROM pracownicy  
WHERE etat = 'ASYSTENT';
```

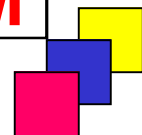


Zapytania z wieloma grupami (1)

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

```
SELECT id_zesp, SUM(placa_pod) FROM pracownicy  
GROUP BY id_zesp;
```

| NAZWISKO | ... | ID_Z | PLACA | |
|------------|-----|------|-------|-------|
| WEGLARZ | ... | 10 | 1730 | } SUM |
| BLAZEWICZ | ... | 40 | 1350 | |
| SLOWINSKI | ... | 30 | 1070 | } SUM |
| BRZEZINSKI | ... | 20 | 960 | |
| MORZY | ... | 20 | 830 | } SUM |
| KOSZLAJDA | ... | 20 | 590 | |
| ZAKRZEWICZ | ... | 30 | 208 | } SUM |
| BIALY | ... | 30 | 250 | |
| BLAZEWICZ | ... | 40 | 1350 | } SUM |



Zapytania z wieloma grupami (2)

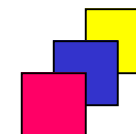
- Wylicz najwyższą płacę wśród pracowników zatrudnionych na każdym etacie

```
SELECT etat, MAX(placa_pod)
FROM pracownicy
GROUP BY etat;
```

Podział grup na podgrupy

- W ramach każdego zespołu dla każdego etatu występującego w zespole oblicz najwyższą płacę

```
SELECT id_zesp, etat, MAX(placa_pod)
FROM pracownicy
GROUP BY id_zesp, etat;
```



Klauzula HAVING

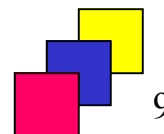
Pozwala na wybór grup spełniających określone warunki, działa dla grup analogicznie jak klauzula WHERE dla pojedynczych krotek

- wyświetl grupy etatowe, których maksymalna płaca podstawowa przekracza 1000 złotych

```
SELECT etat, SUM(placa_pod)
FROM pracownicy
GROUP BY etat
HAVING MAX(placa_pod) > 1000;
```

- wyświetl nazwy etatów i liczbę zatrudnionych na danym etacie, uwzględnij tylko etaty, na których jest zatrudnionych co najmniej 2 pracowników otrzymujących płacę dodatkową

```
SELECT etat, COUNT(*) FROM pracownicy
WHERE placa_dod IS NOT NULL
GROUP BY etat HAVING COUNT(*) >= 2;
```



Sortowanie po grupowaniu

W klauzuli **ORDER BY**, zastosowanej w zapytaniu z grupowaniem, możemy umieścić jedynie atrybuty grupujące i/lub funkcje grupowe

```
SELECT id_zesp, AVG(placa_pod)
FROM pracownicy
GROUP BY id_zesp
HAVING SUM(placa_pod) > 3000
ORDER BY MAX(placa_pod) DESC;
```

Konstrukcje zaawansowane

- Użycie funkcji agregującej jako parametru innej funkcji agregującej; przykład: znajdź maksymalną sumę płac pracowników zatrudnionych na poszczególnych etatach.

```
SELECT MAX(SUM(placa_pod))  
FROM pracownicy  
GROUP BY etat;
```

- Zapytanie z jedną grupą i klauzulą HAVING; przykład: podaj wartość maksymalnej płacy pracowników zespołów 10 i 20, ale tylko wtedy, jeśli liczba pracowników w tych zespołach przekracza 12

```
SELECT MAX(placa_pod) FROM pracownicy  
WHERE id_zesp in (10,20)  
HAVING COUNT(*) > 12;
```

Najczęściej popełniane błędy (1)

- Umieszczenie w klauzuli SELECT zapytania z jedną grupą wyrażenia nie będącego funkcją grupową.

```
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 grupową.

```
SELECT id_zesp, nazwisko, SUM(placa_pod)  
FROM pracownicy GROUP BY id_zesp;
```

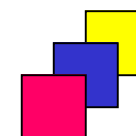
Najczęściej popełniane błędy (2)

- Umieszczenie funkcji grupowej w warunku w klauzuli WHERE.

```
SELECT id_zesp FROM pracownicy  
WHERE COUNT(*) > 3 GROUP BY id_zesp;
```

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

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



Najczęściej popełniane błędy (3)

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

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

Funkcja LISTAGG

- Umożliwia dodanie do rekordu wynikowego dla grupy listy wartości z atrybutów rekordów, należących do grupy.

```
SELECT etat, SUM(placa_pod) AS suma,  
       LISTAGG(nazwisko, ',')  
       WITHIN GROUP (ORDER BY nazwisko) AS pracownicy  
FROM pracownicy GROUP BY etat  
ORDER BY etat;
```

| ETAT | SUMA | PRACOWNICY |
|------------|--------|---|
| ADIUNKT | 1235,5 | KOSZLAJDA, KROLIKOWSKI |
| ASYSTENT | 1770,7 | HAPKE, JEZIERSKI, KONOPKA, MATYSIAK |
| DYREKTOR | 1730 | WEGLARZ |
| PROFESOR | 4210 | BLAZEWICZ, BRZEZINSKI, MORZY, SLOWINSKI |
| SEKRETARKA | 410,2 | MAREK |
| STAZYSTA | 458 | BIALY, ZAKRZEWICZ |

