



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Studia podyplomowe "Inżynieria oprogramowania" współfinansowane przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Projekt "Studia podyplomowe z zakresu wytwarzania oprogramowania oraz zarządzania projektami w firmach informatycznych" realizowany w ramach Programu Operacyjnego Kapitał Ludzki

Konstruowanie Baz Danych

DQL — agregacja danych

Antoni Ligeza

`ligeza@agh.edu.pl`

`http://home.agh.edu.pl/~ligeza`

`http://home.agh.edu.pl/~ligeza/wiki`

Bazy Danych

Wykład p.t.

Agregacja i Grupowanie Danych

Funkcje Agregacji.

Opcje GROUP BY oraz HAVING

Antoni Ligeza

ligeza@agh.edu.pl

<http://galaxy.uci.agh.edu.pl/~ligeza>

Wykorzystano materiały:

[http://www.postgresql.org/docs/8.3/
interactive/index.html](http://www.postgresql.org/docs/8.3/interactive/index.html)

Komendy SQL: SELECT

Command: SELECT

Description: retrieve rows from a table or view

```
SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
      * | expression [ AS output_name ] [, ...]
      [ FROM from_item [, ...] ]
      [ WHERE condition ]
      [ GROUP BY expression [, ...] ]
      [ HAVING condition [, ...] ]
      [ { UNION | INTERSECT | EXCEPT }
[ ALL ] select ]
      [ ORDER BY expression
[ ASC | DESC | USING operator ] [, ...] ]
      [ LIMIT { count | ALL } ]
      [ OFFSET start ]
      [ FOR { UPDATE | SHARE }
[ OF table_name [, ...] ] [ NOWAIT ] [...] ]
```

where from_item can be one of:

```
[ ONLY ] table_name [ * ]
[ [ AS ] alias [ ( column_alias [, ...] ) ] ]
  ( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
  function_name ( [ argument [, ...] ] )
[ AS ] alias [ ( column_alias [, ...]
| column_definition [, ...] ) ]
  function_name ( [ argument [, ...] ] )
AS ( column_definition [, ...] )
  from_item [ NATURAL ] join_type from_item
[ ON join_condition | USING ( join_column [, ...] ) ]
```

Prosta agregacja

SUM – suma.

AVG – średnia.

COUNT (*) – zliczanie rekordów.

```
pracownicy=> SELECT SUM(pobory) AS suma,
                AVG(POBORY) as średnia, COUNT(*) as liczba FROM prac;
  suma      |      średnia      | liczba
-----+-----+-----
 19500.00   | 2437.5000000000000000 |      9
(1 row)
```

```
pracownicy=> SELECT SUM(pobory) AS suma,
                round(AVG(POBORY),2) as średnia,
                COUNT(*) as liczba FROM prac;
  suma      | średnia | liczba
-----+-----+-----
 19500.00   | 2437.50 |      9
(1 row)
```

Operacje agregacji i grupowania

Agregacja:

- łączenie danych z wielu rekordów,
- zwykle znacząco redukuje liczbę rekordów,
- możliwe zliczanie rekordów,
- dotyczy danych numerycznych,
- umożliwia wyliczanie wartości zdefiniowanych funkcji,
- działa również w przypadku wartości NULL,
- tracone są dane *indywidualne*.

Grupowanie:

- możliwe jest tworzenie grup na poziomie:
 - całej tabeli – jedna grupa;
 - grupy w tabeli – jeden stopień grupowania,
 - grupy z podgrupami – grupowanie dwustopniowe,
 - grupowanie hierarchiczne.
- możliwa jest eliminacja grup,
- możliwe jest zliczanie rekordów w grupach,
- możliwe jest porządkowanie wg parametrów grup,
- parametry definiujące grupy mogą być wyświetlane.

Funkcje agregacji

- `AVG (<exp>)` — Średnia z wartości wyrażenia `<exp>`,
- `COUNT (<exp>)` — Liczba wierszy dla których `<exp>` nie jest `NULL`,
- `COUNT (*)` — Liczba wszystkich wierszy,
- `COUNT (DISTINCT (<exp>))` — Liczba wszystkich wierszy dla których wartość `<exp>` jest różna,
- `MAX (<exp>)` — Maksymalna wartość `<exp>`,
- `MIN (<exp>)` — Minimalna wartość `<exp>`,
- `STDDEV (<exp>)` — Odchylenie standardowe dla `<exp>`,
- `SUM (<exp>)` — Suma wartości `<exp>`,
- `VARIANCE (<exp>)` — Wariancja dla `<exp>`.

Typowe przykłady prostej agregacji

```
pracownicy=> SELECT * FROM tpra;
```

id	nazwisko	imie	dzial	stanowisko	pobory
1100	Kowal	Adam	PD303	robotnik	1500.00
110	Kowalik	Artur	PD303	kierownik	1500.00
1110	Kowalewski	Adam	PR202	robotnik	3500.00
101	Kowalczyk	Amadeusz	PK101	kierownik	1000.00
1101	Kowalski	Antoni	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	PK101	robotnik	2500.00
111	Kowalczuk	Adam	PR202	kierownik	2500.00
1010	Kawula	Alojzy	PK101	robotnik	2500.00
102	Janiak	Jerzy	PK101	analitik	6000.00

(9 rows)

```
pracownicy=> SELECT AVG(pobory) FROM tpra;  
          avg
```

```
-----  
2833.3333333333333333  
(1 row)
```

```
pracownicy=> SELECT round(AVG(pobory),2) FROM tpra;  
          round
```

```
-----  
2833.33  
(1 row)
```

Typowe przykłady

Dwa i więcej agregatów:

```
pracownicy=> SELECT sum(pobory), round(AVG(pobory), 2)
              FROM tpra;
```

```
      sum      | round
-----+-----
 25500.00 | 2833.33
(1 row)
```

```
pracownicy=> SELECT sum(pobory),
                  round(AVG(pobory), 2),
                  count(*) FROM tpra;
```

```
      sum      | round | count
-----+-----+-----
 25500.00 | 2833.33 |      9
(1 row)
```

Deklaracja nazw kolumn – agregatów:

```
pracownicy=> SELECT sum(pobory) AS SUMA,
                  round(AVG(pobory), 2) AS SREDNIA,
                  count(*) AS LICZBA FROM tpra;
```

```
      suma      | srednia | liczba
-----+-----+-----
 25500.00 | 2833.33 |      9
(1 row)
```


Użycie COUNT

COUNT (*) – wszystkie rekordy,

COUNT (atrybut) – wszystkie różne od NULL,

COUNT (DISTINCT (atrybut)) wszystkie różne od siebie i od NULL.

```
pracownicy=> select * from tpra;
```

id	nazwisko	imie	dzial	stanowisko	pobory
110	Kowalik	Artur	PD303	kierownik	1500.00
101	Kowalczyk	Amadeusz	PK101	kierownik	1000.00
1101	Kowalski	Antoni	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	PK101	robotnik	2500.00
1010	Kawula	Alojzy	PK101	robotnik	2500.00
102	Janiak	Jerzy	PK101	analitik	6000.00
1100	Kowal		PD303	robotnik	1500.00
1110	Kowalewski		PR202	robotnik	3500.00
111	Kowalczuk		PR202	kierownik	2500.00

(9 rows)

```
pracownicy=> SELECT COUNT(*) FROM tpra;
```

```
count
```

```
-----  
9
```

```
(1 row)
```

```
pracownicy=> SELECT COUNT(imie) FROM tpra;
```

```
count
```

```
-----  
6
```

```
(1 row)
```

```
pracownicy=> SELECT COUNT(DISTINCT(imie)) FROM tpra;
```

```
count
```

```
-----  
5
```

```
(1 row)
```

Agregacja z WHERE

```
pracownicy=> select * from tpra;
```

id	nazwisko	imie	dzial	stanowisko	pobory
110	Kowalik	Artur	PD303	kierownik	1500.00
101	Kowalczyk	Amadeusz	PK101	kierownik	1000.00
1101	Kowalski	Antoni	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	PK101	robotnik	2500.00
1010	Kawula	Alojzy	PK101	robotnik	2500.00
102	Janiak	Jerzy	PK101	analitik	6000.00
1100	Kowal		PD303	robotnik	1500.00
1110	Kowalewski		PR202	robotnik	3500.00
111	Kowalczuk		PR202	kierownik	2500.00

(9 rows)

```
SELECT sum(pobory) AS SUMA,
       round(AVG(pobory),2) AS SREDNIA,
       count(*) AS LICZBA FROM tpra
       WHERE stanowisko <> 'robotnik';
```

suma	srednia	liczba
11000.00	2750.00	4

(1 row)

```
pracownicy=> SELECT sum(pobory) AS SUMA,
       round(AVG(pobory),2) AS SREDNIA,
       count(*) AS LICZBA FROM tpra
       WHERE dzial='PD303';
```

suma	srednia	liczba
7500.00	2500.00	3

(1 row)

Grupowanie – GROUP BY

GROUP BY <exp> — grupowanie jednostopniowe

```
pracownicy=> SELECT * FROM tpra;
```

id	nazwisko	imie	dzial	stanowisko	pobory
110	Kowalik	Artur	PD303	kierownik	1500.00
101	Kowalczyk	Amadeusz	PK101	kierownik	1000.00
1101	Kowalski	Antoni	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	PK101	robotnik	2500.00
1010	Kawula	Alojzy	PK101	robotnik	2500.00
102	Janiak	Jerzy	PK101	analitik	6000.00
1100	Kowal		PD303	robotnik	1500.00
1110	Kowalewski		PR202	robotnik	3500.00
111	Kowalczuk		PR202	kierownik	2500.00

(9 rows)

Tworzymy grupę (np. pracownicy z jednego działu) za pomocą WHERE:

```
pracownicy=> SELECT SUM(pobory), ROUND(AVG(pobory),2),
COUNT(*) FROM tpra WHERE dzial='PK101';
```

sum	round	count
12000.00	3000.00	4

(1 row)

Trzy grupy – trzy działy

```
pracownicy=> SELECT DISTINCT dzial FROM tpra;
```

```
  dzial
```

```
-----
```

```
PD303
```

```
PK101
```

```
PR202
```

```
(3 rows)
```

```
pracownicy=> SELECT SUM(pobory), ROUND(AVG(pobory),2),
                COUNT(*) FROM tpra WHERE dzial='PK101';
```

```
  sum      | round  | count
```

```
-----+-----+-----
```

```
12000.00 | 3000.00 |      4
```

```
(1 row)
```

```
pracownicy=> SELECT SUM(pobory), ROUND(AVG(pobory),2),
                COUNT(*) FROM tpra WHERE dzial='PR202';
```

```
  sum      | round  | count
```

```
-----+-----+-----
```

```
6000.00 | 3000.00 |      2
```

```
(1 row)
```

```
pracownicy=> SELECT SUM(pobory), ROUND(AVG(pobory),2),
                COUNT(*) FROM tpra WHERE dzial='PD303';
```

```
  sum      | round  | count
```

```
-----+-----+-----
```

```
7500.00 | 2500.00 |      3
```

```
(1 row)
```

Jawne grupowanie

```
pracownicy=> SELECT SUM(pobory), ROUND(AVG(pobory),2),  
                  COUNT(*) FROM tpra GROUP BY dzial;
```

sum	round	count
12000.00	3000.00	4
6000.00	3000.00	2
7500.00	2500.00	3

(3 rows)

Można tu podać parametr grupowania — jest stały w grupie.

```
pracownicy=> SELECT dzial,  
                  SUM(pobory),  
                  ROUND(AVG(pobory),2),  
                  COUNT(*)  
                  FROM tpra  
                  GROUP BY dzial;
```

dzial	sum	round	count
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2
PD303	7500.00	2500.00	3

(3 rows)

Grupowanie z ORDER BY

```
pracownicy=> SELECT dzial, SUM(pobory),  
              ROUND(AVG(pobory),2), COUNT(*)  
              FROM tpra GROUP BY dzial;
```

dzial	sum	round	count
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2
PD303	7500.00	2500.00	3

(3 rows)

```
pracownicy=> SELECT dzial, SUM(pobory),  
              ROUND(AVG(pobory),2), COUNT(*)  
              FROM tpra GROUP BY dzial ORDER BY dzial;
```

dzial	sum	round	count
PD303	7500.00	2500.00	3
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2

(3 rows)

```
pracownicy=> SELECT dzial, SUM(pobory),  
              ROUND(AVG(pobory),2), COUNT(*)  
              FROM tpra GROUP BY dzial  
              ORDER BY ROUND(AVG(pobory),2);
```

dzial	sum	round	count
PD303	7500.00	2500.00	3
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2

(3 rows)

Grupowanie z ORDER BY; nazwy i pozycje kolumn

```
pracownicy=> SELECT dzial,
                SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba
                FROM tpra
                GROUP BY dzial
                ORDER BY srednia,suma;
```

dzial	suma	srednia	liczba
PD303	7500.00	2500.00	3
PR202	6000.00	3000.00	2
PK101	12000.00	3000.00	4

(3 rows)

```
pracownicy=> SELECT dzial,
                SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra
                GROUP BY dzial
                ORDER BY 3 DESC, 2 ASC;
```

dzial	suma	srednia	liczba
PR202	6000.00	3000.00	2
PK101	12000.00	3000.00	4
PD303	7500.00	2500.00	3

(3 rows)

HAVING – selekcja grup

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra
                GROUP BY dzial;
```

dzial	suma	srednia	liczba
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2
PD303	7500.00	2500.00	3

(3 rows)

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra
                GROUP BY dzial
                HAVING COUNT(*) > 2;
```

dzial	suma	srednia	liczba
PK101	12000.00	3000.00	4
PD303	7500.00	2500.00	3

(2 rows)

HAVING a WHERE

WHERE — działa na poziomie pojedynczych rekordów,

HAVING — działa na poziomie agregatów (grup).

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra
                GROUP BY dzial;
```

dzial	suma	srednia	liczba
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2
PD303	7500.00	2500.00	3

(3 rows)

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra
                WHERE stanowisko <> 'kierownik'
                GROUP BY dzial;
```

dzial	suma	srednia	liczba
PK101	11000.00	3666.67	3
PR202	3500.00	3500.00	1
PD303	6000.00	3000.00	2

(3 rows)

HAVING a WHERE

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra GROUP BY dzial
                HAVING AVG(pobory) > 2500;
```

dzial	suma	srednia	liczba
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2

(2 rows)

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba FROM tpra
                GROUP BY dzial
                HAVING AVG(pobory) > 3000;
```

dzial	suma	srednia	liczba
-------	------	---------	--------

(0 rows)

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba
                FROM tpra
                WHERE stanowisko <> 'kierownik'
                GROUP BY dzial
                HAVING AVG(pobory) > 3000;
```

dzial	suma	srednia	liczba
PK101	11000.00	3666.67	3
PR202	3500.00	3500.00	1

(2 rows)

Grupowanie dwustopniowe – wstęp

```
pracownicy=> SELECT dzial, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba
                FROM tpra
                GROUP BY dzial;
```

dzial	suma	srednia	liczba
PK101	12000.00	3000.00	4
PR202	6000.00	3000.00	2
PD303	7500.00	2500.00	3

(3 rows)

```
pracownicy=> SELECT stanowisko, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia,
                COUNT(*) AS liczba
                FROM tpra
                GROUP BY stanowisko;
```

stanowisko	suma	srednia	liczba
robotnik	14500.00	2900.00	5
kierownik	5000.00	1666.67	3
analitik	6000.00	6000.00	1

(3 rows)

Grupowanie dwustopniowe

```
pracownicy=> SELECT dzial, stanowisko, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia, COUNT(*) AS liczba
                FROM tpra
                GROUP BY dzial, stanowisko;
```

dzial	stanowisko	suma	srednia	liczba
PR202	robotnik	3500.00	3500.00	1
PK101	analitik	6000.00	6000.00	1
PK101	kierownik	1000.00	1000.00	1
PD303	kierownik	1500.00	1500.00	1
PK101	robotnik	5000.00	2500.00	2
PD303	robotnik	6000.00	3000.00	2
PR202	kierownik	2500.00	2500.00	1

(7 rows)

```
pracownicy=> SELECT dzial, stanowisko, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia, COUNT(*) AS liczba
                FROM tpra
                GROUP BY stanowisko,dzial;
```

dzial	stanowisko	suma	srednia	liczba
PR202	robotnik	3500.00	3500.00	1
PK101	analitik	6000.00	6000.00	1
PK101	kierownik	1000.00	1000.00	1
PD303	robotnik	6000.00	3000.00	2
PR202	kierownik	2500.00	2500.00	1
PD303	kierownik	1500.00	1500.00	1
PK101	robotnik	5000.00	2500.00	2

(7 rows)

Grupowanie dwustopniowe z odrzucaniem grup jednostkowych

```
pracownicy=> SELECT dzial, stanowisko, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia, COUNT(*) AS liczba
                FROM tpra
                GROUP BY dzial, stanowisko
                HAVING COUNT(*) > 1;
```

dzial	stanowisko	suma	srednia	liczba
PK101	robotnik	5000.00	2500.00	2
PD303	robotnik	6000.00	3000.00	2

(2 rows)

```
pracownicy=> SELECT dzial, stanowisko, SUM(pobory) AS suma,
                ROUND(AVG(pobory),2) AS srednia, COUNT(*) AS liczba
                FROM tpra
                GROUP BY stanowisko, dzial
                HAVING COUNT(*) > 1;
```

dzial	stanowisko	suma	srednia	liczba
PD303	robotnik	6000.00	3000.00	2
PK101	robotnik	5000.00	2500.00	2

(2 rows)

GROUP BY + NULL

```
pracownicy=> SELECT * FROM tpra;
```

id	nazwisko	imie	dzial	stanowisko	pobory
110	Kowalik	Artur	PD303	kierownik	1500.00
101	Kowalczyk	Amadeusz	PK101	kierownik	1000.00
1101	Kowalski	Antoni	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	PK101	robotnik	2500.00
1010	Kawula	Alojzy	PK101	robotnik	2500.00
102	Janiak	Jerzy	PK101	analitik	6000.00
1100	Kowal		PD303	robotnik	1500.00
1110	Kowalewski		PR202	robotnik	3500.00
111	Kowalczuk		PR202	kierownik	2500.00

(9 rows)

```
pracownicy=> SELECT imie, count(*)
              FROM tpra
              GROUP BY imie;
```

imie	count
	3
Alojzy	2
Jerzy	1
Antoni	1
Artur	1
Amadeusz	1

(6 rows)



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



**Studia podyplomowe "Inżynieria oprogramowania"
współfinansowane przez Unię Europejską w ramach
Europejskiego Funduszu Społecznego**

Projekt "Studia podyplomowe z zakresu wytwarzania
oprogramowania oraz zarządzania projektami w firmach
informatycznych" realizowany w ramach
Programu Operacyjnego Kapitał Ludzki