



**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

## Podstawy użytkowania instrukcji SELECT

Antoni Ligeza

`ligeza@agh.edu.pl`

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

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

---

## Komendy SQL: SELECT

---

pracownicy=> \h select

Command: SELECT

Description: retrieve rows from a table or view

Syntax:

```
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 [, ...] ) ]
```

---

## SELECT – przykłady

---

```
pracownicy=> select 2+3;
?column?
```

```
-----
          5
(1 row)
```

```
SELECT *
FROM prac;
```

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

id_prac	nazwisko	imie	data_ur	dzial	stanowisko	pobory
1100	Kowal	Adam	1989-12-15	PD303	robotnik	1500.00
110	Kowalik	Artur	1998-12-13	PD303	kierownik	1500.00
1110	Kowalewski	Adam	1989-11-15	PR202	robotnik	3500.00
101	Kowalczyk	Amadeusz	1998-12-17	PK101	kierownik	1000.00
1101	Kowalski	Antoni	1999-12-15	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	1998-11-15	PK101	robotnik	2500.00
111	Kowalczuk	Adam	1998-11-12	PR202	kierownik	2500.00
1010	Kawula	Alojzy	1998-11-15	PK101	robotnik	2500.00

(8 rows)

---

## SELECT – przykłady

---

```
SELECT imie, nazwisko, stanowisko, pobory
FROM prac
WHERE pobory > 1500;
```

```
  imie | nazwisko | stanowisko | pobory
-----+-----+-----+-----
Adam   | Kowalewski | robotnik   | 3500.00
Antoni | Kowalski   | robotnik   | 4500.00
Alojzy | Kowalowski | robotnik   | 2500.00
Adam   | Kowalczuk  | kierownik  | 2500.00
Alojzy | Kawula     | robotnik   | 2500.00
(5 rows)
```

**SELECT: wybór i kolejność atrybutów**

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

id_prac	nazwisko	imie	data_ur	dzial	stanowisko	pobory
1100	Kowal	Adam	1989-12-15	PD303	robotnik	1500.00
110	Kowalik	Artur	1998-12-13	PD303	kierownik	1500.00
1110	Kowalewski	Adam	1989-11-15	PR202	robotnik	3500.00
101	Kowalczyk	Amadeusz	1998-12-17	PK101	kierownik	1000.00
1101	Kowalski	Antoni	1999-12-15	PD303	robotnik	4500.00
1011	Kowalowski	Alojzy	1998-11-15	PK101	robotnik	2500.00
111	Kowalczuk	Adam	1998-11-12	PR202	kierownik	2500.00
1010	Kawula	Alojzy	1998-11-15	PK101	robotnik	2500.00

(8 rows)

```
pracownicy=> \d prac
```

```
Table "public.prac"
  Column      |          Type          | Modifiers
-----+-----+-----
 id_prac      | character(5)           | not null
 nazwisko     | character varying(32) | not null
 imie         | character varying(16) | not null
 data_ur      | date                   | not null
 dzial        | character(5)           | not null
 stanowisko   | character varying(24) |
 pobory       | numeric(8,2)           |
```

Indexes:

```
"prac_pk" PRIMARY KEY, btree (id_prac)
```

Foreign-key constraints:

```
"prac_fk" FOREIGN KEY (dzial) REFERENCES dzial(id_dzial)
ON UPDATE CASCADE
```

```
pracownicy=> SELECT imie, nazwisko, stanowisko, pobory
```

```
pracownicy-> FROM prac;
```

imie	nazwisko	stanowisko	pobory
Adam	Kowal	robotnik	1500.00
Artur	Kowalik	kierownik	1500.00
Adam	Kowalewski	robotnik	3500.00
Amadeusz	Kowalczyk	kierownik	1000.00
Antoni	Kowalski	robotnik	4500.00
Alojzy	Kowalowski	robotnik	2500.00
Adam	Kowalczuk	kierownik	2500.00
Alojzy	Kawula	robotnik	2500.00

## Wyrażenia obliczalne w liście przecinkowej

W wyświetlanej liście wartości można:

- użyć gwiazdki (zastępuje wszystkie pola tablicy),
- ograniczyć listę wypisywanych pól,
- zmienić dowolnie ich kolejność,
- wykonać obliczenia (pola obliczalne),
- przypisać nazwy polom,
- połączyć zawartość pól tekstowych (||).

```
pracownicy=> SELECT imie || ' ' || nazwisko as Pracownik,
pracownicy-> ' pracuje jako ',stanowisko, pobory*0.33 as Premia
pracownicy-> FROM prac;
```

pracownik	?column?	stanowisko	premia
Adam Kowal	pracuje jako	robotnik	495.0000
Artur Kowalik	pracuje jako	kierownik	495.0000
Adam Kowalewski	pracuje jako	robotnik	1155.0000
Amadeusz Kowalczyk	pracuje jako	kierownik	330.0000
Antoni Kowalski	pracuje jako	robotnik	1485.0000
Alojzy Kowalowski	pracuje jako	robotnik	825.0000
Adam Kowalczuk	pracuje jako	kierownik	825.0000
Alojzy Kawula	pracuje jako	robotnik	825.0000

(8 rows)

Nazwy pól występują po słowie AS w cudzysłowach (podwójnych) lub bez.

```
pobory*0.33 AS premia
pobory*0.33 AS "premia"
```

---

## DISTINCT

---

DISTINCT – eliminacja duplikatów; DISTINCT stosuje się do całego wiersza.

```
pracownicy=> SELECT stanowisko from prac;
  stanowisko
```

-----

```
robotnik
kierownik
robotnik
kierownik
robotnik
robotnik
kierownik
robotnik
```

(8 rows)

```
pracownicy=> SELECT DISTINCT stanowisko from prac;
  stanowisko
```

-----

```
kierownik
robotnik
```

DISTINCT ON(<expression>) – różne/pierwsze ze względu na wyrażenie; użycie bez ORDER BY może prowadzić do przypadkowych rezultatów.

```
SELECT DISTINCT ON (location) location, time, report
  FROM weather_reports
  ORDER BY location, time DESC;
```

---

## LIMIT, OFFSET

---

LIMIT <N> – ograniczenie liczby wyświetlanych wierszy.

```
pracownicy=> SELECT nazwisko, stanowisko, pobory FROM prac;
```

```
  nazwisko | stanowisko | pobory
```

```
-----+-----+-----  
Kowal      | robotnik   | 1500.00  
Kowalik    | kierownik  | 1500.00  
Kowalewski | robotnik   | 3500.00  
Kowalczyk  | kierownik  | 1000.00  
Kowalski   | robotnik   | 4500.00  
Kowalowski | robotnik   | 2500.00  
Kowalczuk  | kierownik  | 2500.00  
Kawula     | robotnik   | 2500.00
```

(8 rows)

```
pracownicy=> SELECT nazwisko, stanowisko, pobory FROM prac LIMIT 5;
```

```
  nazwisko | stanowisko | pobory
```

```
-----+-----+-----  
Kowal      | robotnik   | 1500.00  
Kowalik    | kierownik  | 1500.00  
Kowalewski | robotnik   | 3500.00  
Kowalczyk  | kierownik  | 1000.00  
Kowalski   | robotnik   | 4500.00
```

(5 rows)

OFFSET <N> – pominięcie <N> wierszy na początku.



---

```
pracownicy=> SELECT nazwisko, stanowisko, pobory
              FROM prac LIMIT 5;
```

nazwisko	stanowisko	pobory
Kowal	robotnik	1500.00
Kowalik	kierownik	1500.00
Kowalewski	robotnik	3500.00
Kowalczyk	kierownik	1000.00
Kowalski	robotnik	4500.00

(5 rows)

```
pracownicy=> SELECT nazwisko, stanowisko, pobory
              FROM prac LIMIT 5 OFFSET 2;
```

nazwisko	stanowisko	pobory
Kowalewski	robotnik	3500.00
Kowalczyk	kierownik	1000.00
Kowalski	robotnik	4500.00
Kowalowski	robotnik	2500.00
Kowalczuk	kierownik	2500.00

(5 rows)

```
pracownicy=> SELECT nazwisko, stanowisko, pobory
              FROM prac ORDER BY pobory LIMIT 5 OFFSET 2;
```

nazwisko	stanowisko	pobory
Kowal	robotnik	1500.00
Kawula	robotnik	2500.00
Kowalowski	robotnik	2500.00
Kowalczuk	kierownik	2500.00
Kowalewski	robotnik	3500.00

(5 rows)

## WHERE

WHERE <logical\_expression> – po WHERE specyfikuje się kryterium wyboru; musi to być formuła mająca wartość logiczną.

```
pracownicy=> SELECT nazwisko, pobory FROM prac
              WHERE stanowisko='robotnik';
```

nazwisko	pobory
Kowal	1500.00
Kowalewski	3500.00
Kowalski	4500.00
Kowalowski	2500.00
Kawula	2500.00

```
pracownicy=> SELECT nazwisko, stanowisko, pobory
              FROM prac WHERE pobory > 1500;
```

nazwisko	stanowisko	pobory
Kowalewski	robotnik	3500.00
Kowalski	robotnik	4500.00
Kowalowski	robotnik	2500.00
Kowalczyk	kierownik	2500.00
Kawula	robotnik	2500.00

(5 rows)

```
pracownicy=> SELECT nazwisko, imie FROM prac
              WHERE data_ur > '1998-12-15';
```

nazwisko	imie
Kowalczyk	Amadeusz
Kowalski	Antoni

(2 rows)

```
pracownicy=> SELECT nazwisko, imie from PRAC
              WHERE nazwisko > 'Kowal';
```

nazwisko	imie
Kowalik	Artur
Kowalewski	Adam
Kowalczyk	Amadeusz
Kowalski	Antoni
Kowalowski	Alojzy
Kowalczuk	Adam

(6 rows)

## Liczby, łańcuchy i daty

Liczby: ze znakiem lub bez, kropka dziesiętna lub integer; możliwa notacja wykładnicza.

Łańcuchy: w pojedynczych cudzysłowach (apostrofy). Rozróżnianie dużych i małych liter. Uwaga na spacje (i inne białe znaki).

Daty: w pojedynczych cudzysłowach (apostrofy). Uwaga na separator rok-miesiąc, miesiąc-dzień. Uwaga na kolejność (rok, miesiąc, dzień). Uwaga na spacje (i inne białe znaki).

```
pracownicy=> SELECT nazwisko, imie from PRAC
  WHERE nazwisko='kowal';
nazwisko | imie
-----+-----
(0 rows)
```

```
pracownicy=> SELECT nazwisko, imie from PRAC
  WHERE nazwisko='Kowal';
nazwisko | imie
-----+-----
Kowal    | Adam
(1 row)
```

```
pracownicy=> SELECT nazwisko, imie from PRAC
  WHERE nazwisko=' Kowal';
nazwisko | imie
-----+-----
(0 rows)
```

```
pracownicy=> SELECT nazwisko, imie from PRAC
  WHERE nazwisko=' Kowal ';
nazwisko | imie
-----+-----
(0 rows)
```

```
pracownicy=> SELECT nazwisko, imie, data_ur
```

```

FROM prac WHERE data_ur=1989-12-15;
nazwisko | imie | data_ur
-----+-----+-----
(0 rows)

```

```

pracownicy=> SELECT nazwisko, imie, data_ur
FROM prac WHERE data_ur='1989-12-15';
nazwisko | imie | data_ur
-----+-----+-----
Kowal    | Adam | 1989-12-15
(1 row)

```

```

pracownicy=> SELECT nazwisko, imie, data_ur
FROM prac WHERE data_ur='1989/12/15';
nazwisko | imie | data_ur
-----+-----+-----
Kowal    | Adam | 1989-12-15
(1 row)

```

```

pracownicy=> SELECT nazwisko, imie, data_ur
FROM prac WHERE data_ur='1989/12/15 ';
nazwisko | imie | data_ur
-----+-----+-----
Kowal    | Adam | 1989-12-15
(1 row)

```

```

pracownicy=> SELECT nazwisko, imie, data_ur
FROM prac WHERE data_ur=' 1989/12/15 ';
nazwisko | imie | data_ur
-----+-----+-----
Kowal    | Adam | 1989-12-15
(1 row)

```

```

pracownicy=> SELECT nazwisko, imie, data_ur
FROM prac WHERE data_ur='1989/15/12';
ERROR: date/time field value out of range: "1989/15/12"

```

HINT: Perhaps you need a different "datestyle" setting.

Pomocne funkcje:

`ceil(<number>)` – zaokrąglenie w górę.

`floor(<number>)` – zaokrąglenie w dół.

`round(<number>, <places>)` – zaokrąglenie do <places> po przecinku.

`trunc(<number>, <places>)` – obcinanie do <places> po przecinku.

`current_date()` – bieżąca data.

`date_part(<part>, <date>)` – część daty.

`extract(<part> FROM <date>)` – część daty.

`CAST(<arg> AS <type>)` – zmiana typu.

---

```
pracownicy=> SELECT nazwisko, pobory,
               ceil(pobory/3) FROM prac;
nazwisko  | pobory  | ceil
-----+-----+-----
Kowal     | 1500.00 | 500
Kowalik   | 1500.00 | 500
Kowalewski | 3500.00 | 1167
Kowalczyk | 1000.00 | 334
Kowalski  | 4500.00 | 1500
Kowalowski | 2500.00 | 834
Kowalczuk | 2500.00 | 834
Kawula    | 2500.00 | 834
(8 rows)
```

```
pracownicy=> SELECT nazwisko, pobory,
               floor(pobory/3) FROM prac;
nazwisko  | pobory  | floor
-----+-----+-----
Kowal     | 1500.00 | 500
Kowalik   | 1500.00 | 500
Kowalewski | 3500.00 | 1166
Kowalczyk | 1000.00 | 333
Kowalski  | 4500.00 | 1500
Kowalowski | 2500.00 | 833
Kowalczuk | 2500.00 | 833
Kawula    | 2500.00 | 833
(8 rows)
```

```
pracownicy=> SELECT nazwisko, pobory,
               round(pobory/3,3) FROM prac;
nazwisko  | pobory  | round
-----+-----+-----
Kowal     | 1500.00 | 500.000
Kowalik   | 1500.00 | 500.000
Kowalewski | 3500.00 | 1166.667
```

Kowalczyk		1000.00		333.333
Kowalski		4500.00		1500.000
Kowalowski		2500.00		833.333
Kowalczuk		2500.00		833.333
Kawula		2500.00		833.333

(8 rows)

```
pracownicy=> SELECT nazwisko, pobory,
trunc(pobory/3,3) FROM prac;
```

nazwisko		pobory		trunc
-----+-----+				
Kowal		1500.00		500.000
Kowalik		1500.00		500.000
Kowalewski		3500.00		1166.666
Kowalczyk		1000.00		333.333
Kowalski		4500.00		1500.000
Kowalowski		2500.00		833.333
Kowalczuk		2500.00		833.333
Kawula		2500.00		833.333

(8 rows)

```
pracownicy=> SELECT nazwisko,
CAST(pobory AS INTEGER) FROM prac;
```

nazwisko		pobory
-----+-----		
Kowal		1500
Kowalik		1500
Kowalewski		3500
Kowalczyk		1000
Kowalski		4500
Kowalowski		2500
Kowalczuk		2500
Kawula		2500

(8 rows)



## LIKE w SELECT

---

LIKE – wyszukiwanie przybliżone.

% – dowolny ciąg znaków.

\_ – pojedynczy znak. ILIKE – nie rozróżnia dużych/małych liter.

```
string LIKE pattern [ESCAPE escape-character]
string NOT LIKE pattern [ESCAPE escape-character]
```

```
'abc' LIKE 'abc'      true
'abc' LIKE 'a%'      true
'abc' LIKE '_b_'     true
'abc' LIKE 'c'       false
```

SIMILAR TO - rozszerzona składnia LIKE. | – alternatywa symboli, () – jawne grupowanie, \* – powtórzenia (0 możliwe), + – powtórzenia (od 1).

```
string SIMILAR TO pattern [ESCAPE escape-character]
string NOT SIMILAR TO pattern [ESCAPE escape-character]
```

```
'abc' SIMILAR TO 'abc'      true
'abc' SIMILAR TO 'a'        false
'abc' SIMILAR TO '%(b|d)%' true
'abc' SIMILAR TO '(b|c)%'   false
```

```
pracownicy=> SELECT nazwisko FROM prac
              WHERE nazwisko LIKE '%w%l%w%';
              nazwisko
```

```
-----
Kowalewski
Kowalowski
(2 rows)
```

```
pracownicy=> SELECT nazwisko FROM prac
              WHERE nazwisko LIKE 'K%ski';
              nazwisko
```

```
-----
```

Kowalewski  
Kowalski  
Kowalowski

```
pracownicy=> SELECT nazwisko FROM prac
              WHERE nazwisko LIKE 'Kowal_wski';
              nazwisko
```

-----

Kowalewski  
Kowalowski  
(2 rows)

---

```
pracownicy=> SELECT nazwisko FROM prac
              WHERE nazwisko LIKE '%w%l%w%';
              nazwisko
```

```
-----
Kowalewski
Kowalowski
(2 rows)
```

---

## BETWEEN ... AND ...

---

BETWEEN ... AND ... – pomiędzy dwoma wartościami.

```
pracownicy=> SELECT nazwisko, pobory FROM PRAC
              WHERE pobory BETWEEN 1500 AND 2500;
```

nazwisko	pobory
Kowal	1500.00
Kowalik	1500.00
Kowalowski	2500.00
Kowalczyk	2500.00
Kawula	2500.00

(5 rows)

```
pracownicy=> SELECT nazwisko, pobory FROM PRAC
              WHERE nazwisko BETWEEN 'Ka' AND 'Kowalczyk';
```

nazwisko	pobory
Kowal	1500.00
Kowalczyk	1000.00
Kowalczyk	2500.00
Kawula	2500.00

(4 rows)

```
pracownicy=> SELECT nazwisko, data_ur FROM prac
              WHERE data_ur BETWEEN '1998-01-01' AND '1999-12-12';
```

nazwisko	data_ur
Kowalik	1998-12-13
Kowalczyk	1998-12-17
Kowalowski	1998-11-15
Kowalczyk	1998-11-12
Kawula	1998-11-15

(5 rows)



## IN

---

IN – należy do listy.

```
pracownicy=> SELECT nazwisko, dzial, stanowisko
              FROM PRAC WHERE dzial IN ('PR202','PD303','PW101');
```

```
  nazwisko   | dzial | stanowisko
```

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

```
Kowal       | PD303 | robotnik
Kowalik     | PD303 | kierownik
Kowalewski  | PR202 | robotnik
Kowalski    | PD303 | robotnik
Kowalczuk   | PR202 | kierownik
```

```
(5 rows)
```

## IS NULL, IS NOT NULL

---

IS NULL – wykrywanie pól pustych.

IS NOT NULL – wykrywanie pól wypełnionych.

---

```
pracownicy=> SELECT nazwisko, stanowisko FROM PRAC;
```

```
  nazwisko | stanowisko
```

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

```
Kowal      | robotnik
Kowalik    | kierownik
Kowalewski | robotnik
Kowalczyk  | kierownik
Kowalski   | robotnik
Kowalowski | robotnik
Kowalczuk  | kierownik
Kawula     | robotnik
kudryk     |
```

```
(9 rows)
```

```
pracownicy=> SELECT nazwisko, stanowisko FROM PRAC WHERE stanowisko IS NULL;
```

```
  nazwisko | stanowisko
```

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

```
kudryk    |
```

```
(1 row)
```



## Proste łączenie tablic

WHERE tablica1.atrybut1=tablica2.atrybut2 – proste łączenie można wykonać poprzez porównanie pól z dwóch tablic. Najczęściej jest to łączenie klucz-klucz obcy.

```
pracownicy=> SELECT id_prac,nazwisko, imie, nazwa,id_dzial
              FROM dzial, prac WHERE dzial.kierownik=prac.id_prac;
```

id_prac	nazwisko	imie	nazwa	id_dzial
110	Kowalik	Artur	Produkcyjny	PD303
101	Kowalczyk	Amadeusz	Projektowy	PK101
111	Kowalczuk	Adam	Promocji	PR202

(3 rows)

id_prac	nazwisko	imie	nazwa	id_dzial
1100	Kowal	Adam	Produkcyjny	PD303
110	Kowalik	Artur	Produkcyjny	PD303
1110	Kowalewski	Adam	Promocji	PR202
101	Kowalczyk	Amadeusz	Projektowy	PK101
1101	Kowalski	Antoni	Produkcyjny	PD303
1011	Kowalowski	Alojzy	Projektowy	PK101
111	Kowalczuk	Adam	Promocji	PR202
1010	Kawula	Alojzy	Projektowy	PK101
1111	kudryk	adam	Produkcyjny	PD303

(9 rows)

Uwaga: bez (sensownego) warunku łączenia powstaje iloczyn kartezjański!

```
pracownicy=> SELECT * FROM dzial, prac;
```

id_dzial	nazwa	lokalizacja	kierownik	id_prac	nazwisko	imie	data_ur
PD303	Produkcyjny	Mysiecko	110	1100	Kowal	Adam	1989-12-15
PD303	Produkcyjny	Mysiecko	110	110	Kowalik	Artur	1998-12-13
PD303	Produkcyjny	Mysiecko	110	1110	Kowalewski	Adam	1989-11-15
PD303	Produkcyjny	Mysiecko	110	101	Kowalczyk	Amadeusz	1998-12-17
PD303	Produkcyjny	Mysiecko	110	1101	Kowalski	Antoni	1999-12-15
PD303	Produkcyjny	Mysiecko	110	1011	Kowalowski	Alojzy	1998-11-15
PD303	Produkcyjny	Mysiecko	110	111	Kowalczuk	Adam	1998-11-12

PD303		Produkcyjny		Mysiecko		110		1010		Kawula		Alojzy		1998-11-15	
PD303		Produkcyjny		Mysiecko		110		1111		kudryk		adam		1999-12-13	
PK101		Projektowy		Mysiecko		101		1100		Kowal		Adam		1989-12-15	
PK101		Projektowy		Mysiecko		101		110		Kowalik		Artur		1998-12-13	
PK101		Projektowy		Mysiecko		101		1110		Kowalewski		Adam		1989-11-15	
PK101		Projektowy		Mysiecko		101		101		Kowalczyk		Amadeusz		1998-12-17	
PK101		Projektowy		Mysiecko		101		1101		Kowalski		Antoni		1999-12-15	
PK101		Projektowy		Mysiecko		101		1011		Kowalowski		Alojzy		1998-11-15	
PK101		Projektowy		Mysiecko		101		111		Kowalczuk		Adam		1998-11-12	
PK101		Projektowy		Mysiecko		101		1010		Kawula		Alojzy		1998-11-15	
PK101		Projektowy		Mysiecko		101		1111		kudryk		adam		1999-12-13	
PR202		Promocji		Mysieoczko		111		1100		Kowal		Adam		1989-12-15	
PR202		Promocji		Mysieoczko		111		110		Kowalik		Artur		1998-12-13	
PR202		Promocji		Mysieoczko		111		1110		Kowalewski		Adam		1989-11-15	
PR202		Promocji		Mysieoczko		111		101		Kowalczyk		Amadeusz		1998-12-17	
PR202		Promocji		Mysieoczko		111		1101		Kowalski		Antoni		1999-12-15	
PR202		Promocji		Mysieoczko		111		1011		Kowalowski		Alojzy		1998-11-15	
PR202		Promocji		Mysieoczko		111		111		Kowalczuk		Adam		1998-11-12	
PR202		Promocji		Mysieoczko		111		1010		Kawula		Alojzy		1998-11-15	
PR202		Promocji		Mysieoczko		111		1111		kudryk		adam		1999-12-13	

(27 rows)

---

## 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)

## ORDER BY

ORDER BY – sortowanie wyników.

```
ORDER BY sort_expression1 [ASC | DESC] [NULLS { FIRST | LAST }]
      [, sort_expression2 [ASC | DESC] [NULLS { FIRST | LAST }] ...]
```

```
pracownicy=> SELECT nazwisko, imie, dzial, stanowisko
              FROM prac ORDER BY nazwisko;
```

nazwisko	imie	dzial	stanowisko
Kawula	Alojzy	PK101	robotnik
Kowal	Adam	PD303	robotnik
Kowalczuk	Adam	PR202	kierownik
Kowalczyk	Amadeusz	PK101	kierownik
Kowalewski	Adam	PR202	robotnik
Kowalik	Artur	PD303	kierownik
Kowalowski	Alojzy	PK101	robotnik
Kowalski	Antoni	PD303	robotnik
kudryk	adam	PD303	

(9 rows)

```
pracownicy=> SELECT nazwisko, imie, dzial, stanowisko
              FROM prac ORDER BY stanowisko;
```

nazwisko	imie	dzial	stanowisko
Kowalczyk	Amadeusz	PK101	kierownik
Kowalczuk	Adam	PR202	kierownik
Kowalik	Artur	PD303	kierownik
Kowalowski	Alojzy	PK101	robotnik
Kawula	Alojzy	PK101	robotnik
Kowal	Adam	PD303	robotnik
Kowalewski	Adam	PR202	robotnik
Kowalski	Antoni	PD303	robotnik
kudryk	adam	PD303	

(9 rows)

```
pracownicy=> SELECT nazwisko, imie, dzial, stanowisko
              FROM prac ORDER BY imie;
```

nazwisko	imie	dzial	stanowisko
kudryk	adam	PD303	
Kowalewski	Adam	PR202	robotnik
Kowal	Adam	PD303	robotnik
Kowalczuk	Adam	PR202	kierownik
Kowalowski	Alojzy	PK101	robotnik
Kawula	Alojzy	PK101	robotnik
Kowalczyk	Amadeusz	PK101	kierownik
Kowalski	Antoni	PD303	robotnik
Kowalik	Artur	PD303	kierownik

(9 rows)

```
pracownicy=> SELECT nazwisko, imie, dzial, stanowisko
              FROM prac ORDER BY nazwisko DESC;
```

nazwisko	imie	dzial	stanowisko
kudryk	adam	PD303	
Kowalski	Antoni	PD303	robotnik
Kowalowski	Alojzy	PK101	robotnik
Kowalik	Artur	PD303	kierownik
Kowalewski	Adam	PR202	robotnik
kowalecki	adam	PD303	
Kowalczyk	Amadeusz	PK101	kierownik
Kowalczuk	Adam	PR202	kierownik
Kowal	Adam	PD303	robotnik
Kawula	Alojzy	PK101	robotnik

(10 rows)

```
pracownicy=> SELECT nazwisko, imie, dzial, stanowisko
              FROM prac ORDER BY stanowisko;
```

nazwisko	imie	dzial	stanowisko
Kowalczyk	Amadeusz	PK101	kierownik
Kowalczuk	Adam	PR202	kierownik

Kowalik		Artur		PD303		kierownik
Kowalowski		Alojzy		PK101		robotnik
Kawula		Alojzy		PK101		robotnik
Kowal		Adam		PD303		robotnik
Kowalewski		Adam		PR202		robotnik
Kowalski		Antoni		PD303		robotnik
kudryk		adam		PD303		
kowalecki		adam		PD303		

(10 rows)

```
pracownicy=> SELECT nazwisko, imie, dzial, stanowisko
```

```
FROM prac ORDER BY stanowisko DESC;
```

```
nazwisko | imie | dzial | stanowisko
```

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

kowalecki		adam		PD303		
kudryk		adam		PD303		
Kowalski		Antoni		PD303		robotnik
Kowal		Adam		PD303		robotnik
Kowalowski		Alojzy		PK101		robotnik
Kowalewski		Adam		PR202		robotnik
Kawula		Alojzy		PK101		robotnik
Kowalik		Artur		PD303		kierownik
Kowalczyk		Amadeusz		PK101		kierownik
Kowalczuk		Adam		PR202		kierownik

(10 rows)

## Spójniki logiczne: AND, OR, NOT

AND – koniunkcja warunków,

OR - dysjunkcja,

NOT – negacja.

```
pracownicy=> SELECT nazwisko, stanowisko, dzial, pobory
FROM prac;
```

nazwisko	stanowisko	dzial	pobory
Kowal	robotnik	PD303	1500.00
Kowalik	kierownik	PD303	1500.00
Kowalewski	robotnik	PR202	3500.00
Kowalczyk	kierownik	PK101	1000.00
Kowalski	robotnik	PD303	4500.00
Kowalowski	robotnik	PK101	2500.00
Kowalczuk	kierownik	PR202	2500.00
Kawula	robotnik	PK101	2500.00
kudryk		PD303	
kowalecki		PD303	

(10 rows)

```
pracownicy=> SELECT nazwisko, stanowisko, dzial, pobory
FROM prac WHERE stanowisko='robotnik' AND pobory>1500;
```

nazwisko	stanowisko	dzial	pobory
Kowalewski	robotnik	PR202	3500.00
Kowalski	robotnik	PD303	4500.00
Kowalowski	robotnik	PK101	2500.00
Kawula	robotnik	PK101	2500.00

(4 rows)

```
pracownicy=> SELECT nazwisko, stanowisko, dzial, pobory FROM prac
WHERE (stanowisko='robotnik' AND pobory>1500) OR stanowisko='kierownik';
```

nazwisko	stanowisko	dzial	pobory
----------	------------	-------	--------

Kowalik		kierownik		PD303		1500.00
Kowalewski		robotnik		PR202		3500.00
Kowalczyk		kierownik		PK101		1000.00
Kowalski		robotnik		PD303		4500.00
Kowalowski		robotnik		PK101		2500.00
Kowalczuk		kierownik		PR202		2500.00
Kawula		robotnik		PK101		2500.00

(7 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