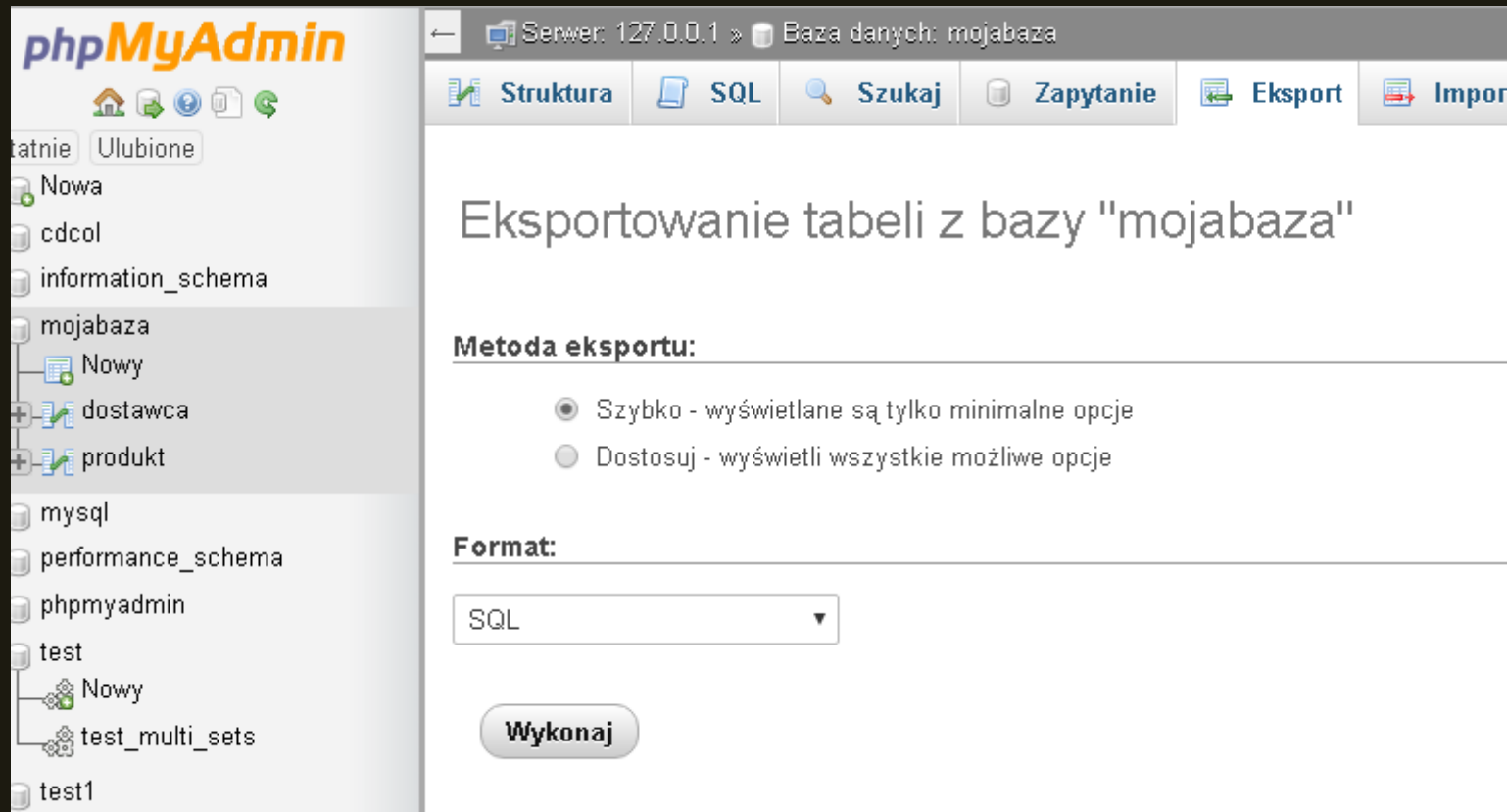


**Pobieranie danych**

**SELECT**

**(+ Backup)**

# Backup – zapis kopi bazy danych (struktura + dane) oraz ich odzysk z pliku...



**Wybór bazy danych -> Export -> Szybko (Format: SQL)-> Wykonaj**

**W razie potrzeby można wybrać metodę: Dostosuj...**

**Metoda eksportu:**

Szybko - wyświetlane są tylko minimalne opcje

Dostosuj - wyświetli wszystkie możliwe opcje

**Tabela(e):**

Zaznacz wszystkie / Odznacz wszystko

dostawca
produkt

**Wyjście:**

Zapisz wynik do pliku

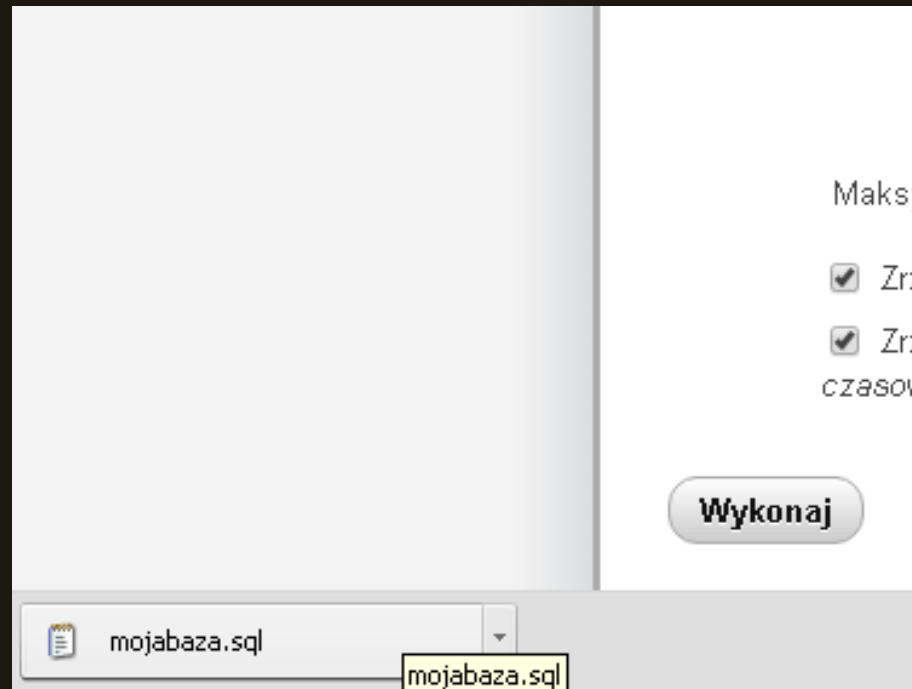
Szablon nazwy pliku:   użyj tego dla przyszłych

Kodowanie znaków pliku:

Kompresja:

**Wówczas można określić szczegółowe parametry eksportu (np. tabele, które chce się eksportować..)**

Po kliknięciu na WYKONAJ plik z kopią bazy danych zostanie  
'ściągnięty' na dysk lokalny:



**UWAGA:** jeśli baza danych zawiera dużo danych plik może sporo  
'wazyć'. W takiej sytuacji można ściągać np. po jednej tabeli, lub w  
parametrach eksportu wybrać opcję kompresji...

Bazy danych SQL Status Użytkownicy Eksport Import Ustaw

## Importowanie do bieżącego serwera

**Plik do importu:**

Plik może być skompresowany (gzip, bzip2, zip) bądź nie.  
Plik skompresowany musi mieć rozszerzenie **.[format].[kompresja]**, np. **.sql.zip**

Wyszukaj w komputerze:  mojabaza.sql (Maksymalny rozmiar: 2,048KB)

Kodowanie znaków pliku:

**Import częściowy:**

Zezwól na przerwanie importu w przypadku, gdy skrypt wykryje zbliżający się koniec limitu czasu. (Może to być spowodowane przez błąd lub może on popsuć transakcje.)

Pomiń tę liczbę zapytań (dla SQL) lub linii (dla innych formatów), począwszy od pierwszego:

**Format:**

**Opcje formatu:**

Tryb zgodności SQL:

Nie używaj AUTO\_INCREMENT dla wartości zerowych

## Odtwarzanie bazy danych z pliku kopi \*.sql

**1.Należy utworzyć 'pustą' bazę danych – po jej utworzeniu należy ją wybrać.**

**2.Wybieramy zakładkę IMPORT.**

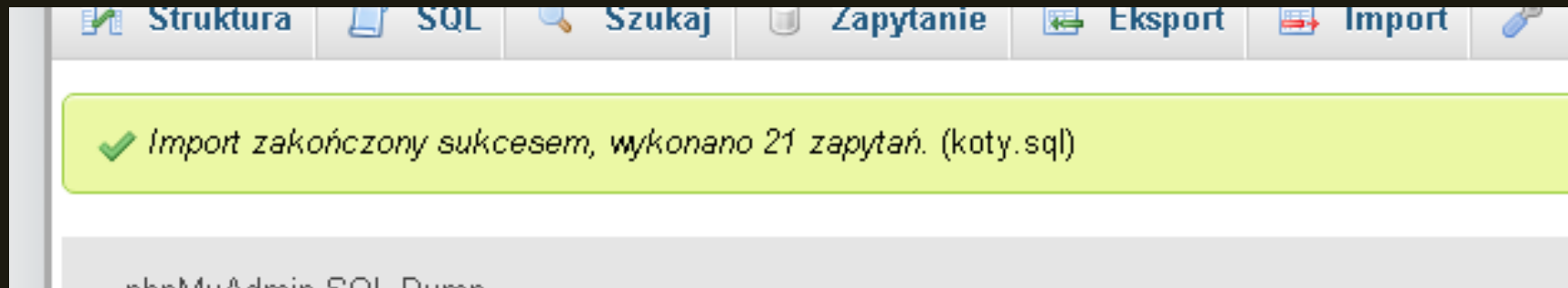
**3.Należy wskazać plik (pliki – po jednym – jeśli jest ich więcej) z kopią.**

**4.Jeśli nie ma potrzeby zmiany parametrów importu klikamy**

**WYKONAJ.**

**5.Jeśli baza jest zapisana w więcej niż jednym pliku wracamy do punktu 2.**

Po poprawnym imporcie otrzymujemy stosowny komunikat oraz w bazie danych jest już nasza importowana baza (lub jej część w przypadku importu częściowego).



**ĆWICZENIE: Wykonaj import bazy danych koty.sql (ze wskazanej przez nauczyciela lokalizacji). W pierwszej kolejności należy stworzyć pustą bazę danych o nazwie wskazanej przez nauczyciela.**

**ĆWICZENIE: Po poprawnym Importie zapisz kopię bazy danych (pod nazwą wskazaną przez nauczyciela).**

**SELECT**



# SKŁADNIA:

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr ...]
[FROM table_references
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
[ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col_name | expr | position}
[ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[PROCEDURE procedure_name(argument_list) ]
[INTO OUTFILE 'file_name' export_options
| INTO DUMPFILE 'file_name'
| INTO var_name [, var_name]]
[FOR UPDATE | LOCK IN SHARE MODE]]
```

**SELECT w przykładach**  
**(nie taki straszny...)**

```
SELECT * FROM users;
```

Pobrane zostaną wszystkie wiersze z tabeli **users**.  
**Gwiazdka** oznacza, że pobieramy wszystkie dane (kolumny) opisujące usera.

ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55
10	Maria	22
11	Franek	44
12	Maja	44

**SELECT imie, wiek FROM users;**

Pobrane zostaną wszystkie wiersze z tabeli **users**.  
Tym razem pobieramy jednak wybrane wiersze opisujące  
USERA: **Imie i Wiek**.

IMIE	WIEK
Jacek	43
Marek	45
Marta	34
Jola	23
Jasio	34
Radek	54
Marek	23
Jan	33
Olek	55
Maria	22
Franek	44
Maja	44

## SELECT DISTINCT imie FROM users;

MySQL domyślnie zwraca wszystkie wartości niezależnie od tego czy są takie same czy nie, jeśli nie chcemy powtarzających się wyników użyjemy wtedy operatora DISTINCT.

BEZ DISTINCT

IMIE
Jacek
Marek
Marta
Jola
Jasio
Radek
Marek
Jan
Olek
Maria
Franek
Maja

Z DISTINCT

IMIE
Jacek
Marek
Marta
Jola
Jasio
Radek
Jan
Olek
Maria
Franek
Maja

**Zobacz, że po użyciu DISTINCT MAREK  
pojawia się już tylko raz...**

## DISTINCT – jeszcze raz...

**SELECT DISTINCT imie, wiek FROM users;**

<b>imie</b>	<b>WIEK</b>
Jacek	43
Marek	45
Marta	34
Jola	23
Jasio	34
Radek	54
Marek	23
Jan	33
Olek	55
Maria	22
Franek	44
Maja	44

**W tym wypadku distinct nie zadziałał jak można by się tego spodziewać – Mamy na liście 2 MARKÓW...**

**...Stało się tak dlatego ponieważ, każdy z Marków ma inny wiek. Wcześniej gdy pobierane były tylko imiona jeden i drugi Marek ‘wyglądali tak samo’...**

**...Teraz gdy dodatkowo pobierany jest wiek jeden Marek różni się troszkę od drugiego Marka.**

**Gdyby zmienić wiek drugiego marka tak by był on taki sam jak tego pierwszego wynik powyższego działania będzie następujący:**

<b>imie</b>	<b>wiek</b>
Jacek	43
Marek	45
Marta	34
Jola	23
Jasio	34
Radek	54
Jan	33
Olek	55
Maria	22
Franek	44
Maja	44

**Jak widać na zdjęciu na liście jest tylko jeden Marek – teraz jeden Marek nie różni się od drugiego więc pokazany został tylko jeden...**

**Marek – 45**

**Marek – 45 (było by powtórzenie – nie będzie widoczny)**

# SELECT id as identyfikator FROM users;

**Aliasy:** Za pomocą słowa **AS** możemy utworzyć alias dla nazw kolumn, tabel.

W ten sposób słowo **id** zostało zastąpione słowem **identyfikator**, aliasów możemy używać także w klauzuli **FROM** dla doraźnej zmiany nazw tabel. Aliasy są pomocne m.in. kiedy w jednym zapytaniu korzysta się z danych pochodzących z różnych tabel – wówczas w dwóch różnych tabelach może pojawić się atrybut o tej samej nazwie – alias pozwala je ponazywać na ‘nowo’.

Uwaga na później: klauzula **GROUP BY** oraz **ORDER BY** jako argument nie może używać wyrazów złożonych, wtedy konieczne należy zastąpić je aliasami.

**UWAGA:** użycie aliasu nie zmienia trwale nazw w bazie danych. Zmiana ta występuje jedynie doraźnie na potrzeby tego jednego zapytania, w którym zostały użyte.

Przykład:

identyfikator
1
2
3
4
5
6
7
8
9
10
11
12

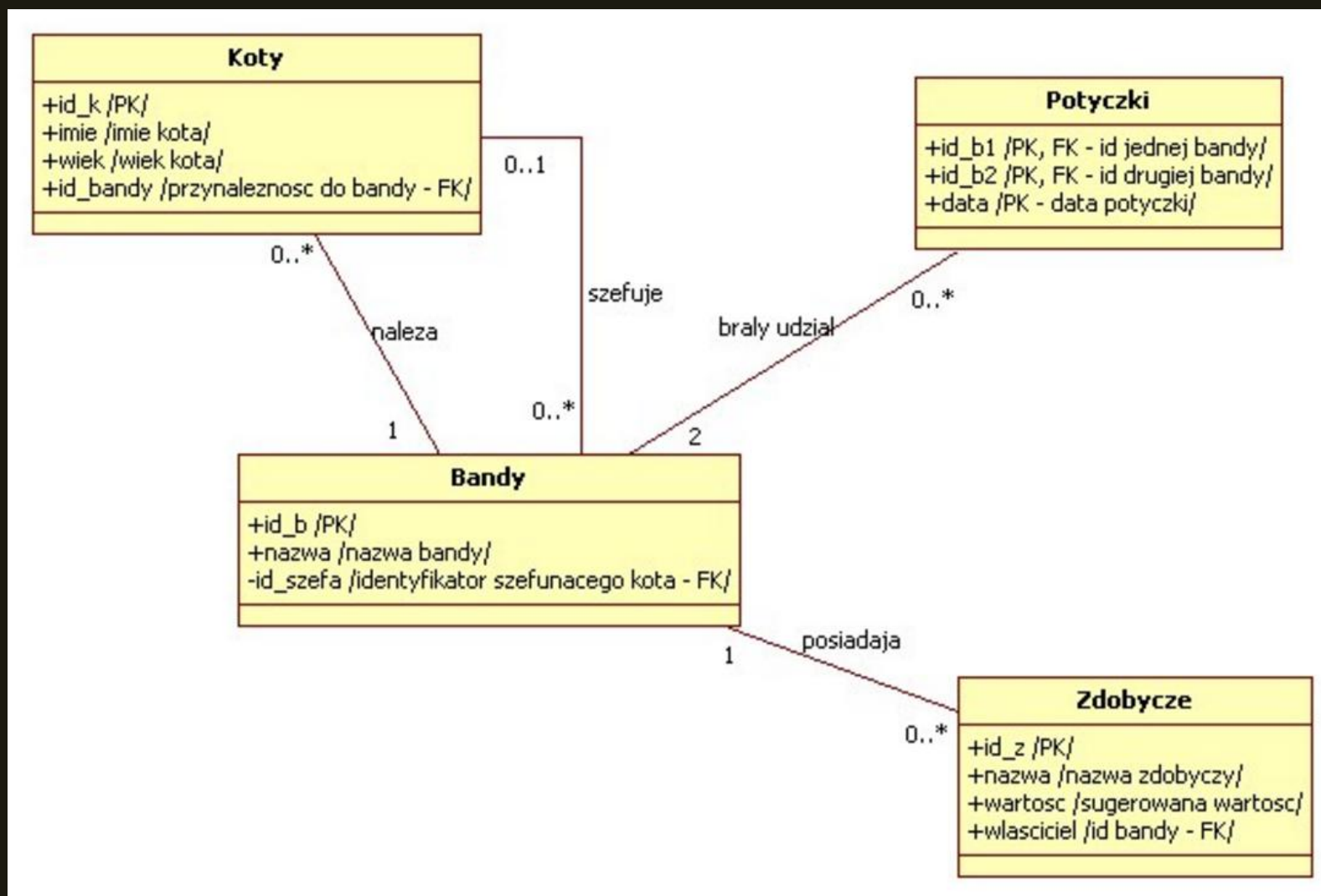


Aliasy raz jeszcze...

```
SELECT id as identyfikator, imie, wiek as ile_lat  
FROM users;
```

identyfikator	imie	ile_lat
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	45
8	Jan	33
9	Olek	55
10	Maria	22
11	Franek	44
12	Maja	44

Pora na kilka ćwiczeń – ćwiczenia dotyczą zaimportowanej bazy KOTÓW. Dla uściślenia poniżej grafika prezentująca strukturę danych w bazie koty (ta sama grafika jest dostępna również pod adresem wskazanym przez nauczyciela).



**ĆWICZENIE: wyświetl wszystkie dane kotów.**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
6	gotfryd	7	1
7	marmelada	2	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
11	zdzisio	9	1
12	mruczus	6	1
13	czarnulka	4	3
14	zbyszko	2	1
15	bury	7	1
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

# SELECT \* FROM koty

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
6	gotfryd	7	1
7	marmelada	2	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
11	zdzisio	9	1
12	mruczus	6	1
13	czarnulka	4	3
14	zbyszko	2	1
15	bury	7	1
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

**ĆWICZENIE: Wyświetl imiona i wiek kotów.**

imie	wiek
gapcio	3
fruzia	12
smykos	4
lady gada	5
myszka	2
gotfryd	7
marmelada	2
leopard	9
szemrak	12
macius	23
zdzisio	9
mruczus	6
czarnulka	4
zbyszko	2
bury	7
zadziora	2
franka	7
malutka	2
milka	3
stasiu	2

## SELECT imie, wiek FROM koty

imie	wiek
gapcio	3
fruzia	12
smykos	4
lady gada	5
myszka	2
gotfryd	7
marmelada	2
leopard	9
szemrak	12
macius	23
zdzisio	9
mruczus	6
czarnulka	4
zbyszko	2
bury	7
zadziora	2
franka	7
malutka	2
milka	3
stasiu	2

ĆWICZENIE: Wyświetl wiek kotów (bez powtórzeń).

wiek
3
12
4
5
2
7
9
23
6

**SELECT DISTINCT wiek FROM koty**

wiek
3
12
4
5
2
7
9
23
6



**ĆWICZENIE:** Wyświetl dane kotów w taki sposób, by zamiast nazwy id\_bandy ponawiał się nagłówek **BANDZIOCHA**.

id_k	imie	wiek	BANDZIOCHA
1	gapcio	3	1
2	fruzia	12	1
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
6	gotfryd	7	1
7	marmelada	2	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
11	zdzisio	9	1
12	mruczus	6	1
13	czarnulka	4	3
14	zbyszko	2	1
15	bury	7	1
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

**SELECT id\_k, imie, wiek, id\_bandy as BANDZIOCHA FROM koty**

id_k	imie	wiek	BANDZIOCHA
1	gapcio	3	1
2	fruzia	12	1
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
6	gotfryd	7	1
7	marmelada	2	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
11	zdzisio	9	1
12	mruczus	6	1
13	czarnulka	4	3
14	zbyszko	2	1
15	bury	7	1
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

# SELECT

**Klauzula WHERE**

**SELECT imie FROM users WHERE id=10;**

Klauzula **WHERE** służy do warunkowego pobierania danych.

Pobierze tylko te wiersze gdzie **id** jest równe **10**

Klauzuli tej można także używać przy zapytaniach typu **UPDATE, DELETE** etc.

ID	IMIE	WIEK
10	Maria	22

**W wyrażeniu WHERE możemy używać różnych operatorów:**

- =** - równy
- <>** - nierówny
- <** - mniejszy od
- >** - większy od
- <=** - mniejszy lub równy
- >=** - większy lub równy
- !** - negacja

Dodatkowo można łączyć warunki za pomocą operatorów:

- AND** - i
- OR** - lub

Możemy także użyć nawiasów w celu ustawienia kolejności wykonywania działań matematycznych (i nie tylko).

## Przykłady:

```
SELECT * FROM users WHERE id <> 10
```

ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55
11	Franek	44
12	Maja	44

```
SELECT * FROM users WHERE id < 10
```

ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55

```
SELECT * FROM `users` WHERE id < 10 AND wiek < 30
```

**UWAGA:** pierwsze użycie operatora AND (szukamy osób, których id jest mniejsze od 10 i wiek jest mniejszy od 30)...

ID	IMIE	WIEK
4	Jola	23
7	Marek	23



ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55
10	Maria	22
11	Franek	44
12	Maja	44

## Jak działają operatory AND i OR (przypomnienie logiki...)

Każdy warunek daje wynik

logiczne TAK (true = 1) lub NIE (false = 0)

**ZAPYTANIE:**        `SELECT * FROM users WHERE id < 10`

**BAZA DANYCH REALIZUJE W TAK:** bierze kolejno wiersze z tabeli users i podstawia wartość pola ID w miejsce warunku...

Czyli weźmie Jacka, Marka, Martę itd... kolejno, podstawia do warunku ich ID i sprawdzi czy warunek jest prawdziwy. Po czym zwróci te wiersze, w których warunek dawał PRAWDĘ.

Dla **Marii**, **Franka** i **Maji**, warunek nie będzie prawdziwy.

Zatem prawdą będzie, że:

Kolejne wartości ID czyli, 1, 2, 3, ... 9 będą mniejsze od 10

$1 < 10 \rightarrow \text{TAK (1)}$

$2 < 10 \rightarrow \text{TAK (1)}$

...

$9 < 10 \rightarrow \text{TAK (1)}$

Ale wartości 10, 11, 12 nie będą mniejsze od 10 czyli

$10 < 10 \rightarrow \text{NIE (0)}$

$11 < 10 \rightarrow \text{NIE (0)}$

$12 < 10 \rightarrow \text{NIE (0)}$

ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55
10	Maria	22
11	Franek	44
12	Maja	44

Co w wypadku operatorów AND i OR...?

AND			OR		
IN	IN	OUT	IN	IN	OUT
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

Czyli zapytanie **SELECT \* FROM `users` WHERE id < 10 AND wiek < 30** oznacza, że baza danych wypisze tylko tych userów, który mają zarówno ID mniejsze od 10 jak również wiek mniejszy od 30...

Więcej wyjaśnień – następny slajd...

AND		
IN	IN	OUT
0	0	0
0	1	0
1	0	0
1	1	1

OR		
IN	IN	OUT
0	0	0
0	1	1
1	0	1
1	1	1

ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55
10	Maria	22
11	Franek	44
12	Maja	44

**SELECT \* FROM `users` WHERE id < 10 AND wiek < 30**

Dla JACKA warunek będzie wyglądał następująco:

1
0  
 ID < 10    AND    WIEK < 30    ->    1 < 10    AND    43 < 30

Zgodnie z tabelą    1 and 0 = 0    czyli JACEK nie będzie wypisany

AND		
IN	IN	OUT
0	0	0
0	1	0
1	0	0
1	1	1

OR		
IN	IN	OUT
0	0	0
0	1	1
1	0	1
1	1	1

**SELECT \* FROM `users` WHERE id < 10 AND  
wiek < 30**

Dla MARTY warunek będzie wyglądał  
następująco:

ID	IMIE	WIEK
1	Jacek	43
2	Marek	45
3	Marta	34
4	Jola	23
5	Jasio	34
6	Radek	54
7	Marek	23
8	Jan	33
9	Olek	55
10	Maria	22
11	Franek	44
12	Maja	44

**ID < 10      AND      WIEK < 30**

->

**3 < 10      AND      34 < 30**

**Zgodnie z tabelą      1 and 0 = 0**

**czyli MARTA nie będzie wypisana**

**Dla JOLI**

**ID < 10      AND      WIEK < 30**

->

**4 < 10      AND      23 < 30**

**Zgodnie z tabelą      1 and 1 = 1**

**czyli JOLA będzie wypisana**

**ĆWICZENIE: Wypisz dane kotów, których wiek jest mniejszy niż 3 lata.**

id_k	imie	wiek	id_bandy
5	myszka	2	2
7	marmelada	2	3
14	zbyszko	2	1
16	zadziora	2	2
18	malutka	2	2
20	stasiu	2	1

**ĆWICZENIE:**

**Wypisz dane kotów, których wiek jest z przedziału  $<3, 5)$  lata.**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
3	smykos	4	2
13	czarnulka	4	3
19	milka	3	3

**SELECT \* FROM koty WHERE wiek < 3**

id_k	imie	wiek	id_bandy
5	myszka	2	2
7	marmelada	2	3
14	zbyszko	2	1
16	zadziora	2	2
18	malutka	2	2
20	stasiu	2	1

**SELECT \* FROM koty WHERE wiek >= 3 and wiek < 5**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
3	smykos	4	2
13	czarnulka	4	3
19	milka	3	3

**ĆWICZENIE:** Wypisz dane kotów, które są starsze niż 5 lat oraz należą do bandy nr 3.

id_k	imie	wiek	id_bandy
17	franka	7	3

**ĆWICZENIE:** Wypisz dane kotów, które są młodsze niż 5 lat lub należą do bandy nr 3.

id_k	imie	wiek	id_bandy
1	gapcio	3	1
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
7	marmelada	2	3
13	czarnulka	4	3
14	zbyszko	2	1
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1



**SELECT \* FROM koty WHERE wiek > 5 and id\_bandy = 3**

id_k	imie	wiek	id_bandy
17	franka	7	3

**SELECT \* FROM koty WHERE wiek < 5 or id\_bandy = 3**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
7	marmelada	2	3
13	czarnulka	4	3
14	zbyszko	2	1
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

**Operatory AND i OR – użycie nawiasów...**

**Użycie nawiasów w warunku działa tak samo jak użycie nawiasów w działaniu matematycznym – jest w stanie zmieniać priorytety wykonywania operacji.**

**Przykład matematyczny:**

$$2+3*3 = 2+9 = 11 \quad \text{ale} \quad (2+3)*3 = 5*3 = 15$$

Dokładnie tak samo działają nawiasy w warunkach:

Dla tego samego kota

**ID\_K < 4 or ID\_BANDY < 2 and WIEK < 10**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
3	smykos	4	2
6	gotfryd	7	1
8	leopard	9	1
11	zdzisio	9	1
12	mruczus	6	1
14	zbyszko	2	1
15	bury	7	1
20	stasiu	2	1

ALE...

**(ID K < 4 or ID BANDY < 2) and WIEK < 10**

.....brak FRUZI...?

id_k	imie	wiek	id_bandy
1	gapcio	3	1
3	smykos	4	2
6	gotfryd	7	1
8	leopard	9	1
11	zdzisio	9	1
12	mruczus	6	1
14	zbyszko	2	1
15	bury	7	1
20	stasiu	2	1

**ĆWICZENIE: Wypisz dane kotów, które należą do bandy nr 1 i są młodsze niż 3 lata lub starsze niż 10 lat.**

id_k	imie	wiek	id_bandy
2	fruzia	12	1
9	szemrak	12	1
10	macius	23	1
14	zbyszko	2	1
20	stasiu	2	1

**ĆWICZENIE: Wypisz dane kotów, które należą do bandy nr 1 i są starsze niż 10 lata lub są młodsze niż 3 lata i należą do dowolnej bandy.**

id_k	imie	wiek	id_bandy
2	fruzia	12	1
5	myszka	2	2
7	marmelada	2	3
9	szemrak	12	1
10	macius	23	1
14	zbyszko	2	1
16	zadziora	2	2
18	malutka	2	2
20	stasiu	2	1

**SELECT \* FROM koty WHERE id\_bandy = 1 and (wiek < 3 or wiek > 10);**

id_k	imie	wiek	id_bandy
2	fruzia	12	1
9	szemrak	12	1
10	macius	23	1
14	zbyszko	2	1
20	stasiu	2	1

**SELECT \* FROM koty WHERE (id\_bandy = 1 and wiek > 10) or wiek < 3;**

id_k	imie	wiek	id_bandy
2	fruzia	12	1
5	myszka	2	2
7	marmelada	2	3
9	szemrak	12	1
10	macius	23	1
14	zbyszko	2	1
16	zadziora	2	2
18	malutka	2	2
20	stasiu	2	1

# SELECT \* FROM users WHERE email IS NULL;

## Operator IS NULL

Jeśli miałeś już do czynienia z innymi językami programowania (choćaby PHP), to możliwe że to co powiem teraz zdziwi cię, ale w MySQL pusta wartość (także 0) nie jest tym samym co NULL . Jeśli chcesz pobrać tylko niewypełnione wartości powinieneś skorzystać z polecenia IS NULL.

ID	IMIE	WIEK	email
1	Jacek	43	NULL
2	Marek	45	e1@op.pl
3	Marta	34	NULL
4	Jola	23	e2@op.pl
5	Jasio	34	NULL
6	Radek	54	e3@op.pl
7	Marek	23	NULL
8	Jan	33	e4@op.pl
9	Olek	55	e5@op.pl
10	Maria	22	NULL
11	Franek	44	NULL
12	Maja	44	NULL

Powyższy przykład wyświetli dane tych userów, którzy nie mają wartości dla pola email (wartość tego pola jest pusta):

ID	IMIE	WIEK	email
1	Jacek	43	NULL
3	Marta	34	NULL
5	Jasio	34	NULL
7	Marek	23	NULL
10	Maria	22	NULL
11	Franek	44	NULL
12	Maja	44	NULL

# SELECT \* WHERE email IS NOT NULL;

## Operator NOT

Operator negacji działa tak, że odwraca wartość logiczną wyrażenia.

Jeśli w wyrażeniu **WHERE email IS NULL** szukaliśmy wszystkich, dla których wartość email była pusta, ....

...to odpowiednio w wyrażeniu **WHERE email IS NOT NULL** szukamy tych, dla których wartość email nie jest pusta...

ID	IMIE	WIEK	email
2	Marek	45	e1@op.pl
4	Jola	23	e2@op.pl
6	Radek	54	e3@op.pl
8	Jan	33	e4@op.pl
9	Olek	55	e5@op.pl

**SELECT \* FROM users WHERE imie LIKE 'Ma%';**

## Operator LIKE

Służy do przeszukiwania pól w poszukiwaniu danego wyrażenia. W operatorze LIKE możemy korzystać z dwóch metaznaków:

- %** - zastępuje dowolny ciąg
- \_** - zastępuje dowolny znak

Zapytanie to będzie pasowało do każdego usera którego imie zaczyna się od 'Ma', czyli będzie pasowało zarówno do Marek jak i Marta.

Operator ten znajduje często zastosowanie przy budowie wszelkiego rodzaju wyszukiwarek.

ID	IMIE	WIEK	email
1	Jacek	43	NULL
2	Marek	45	e1@op.pl
3	Marta	34	NULL
4	Jola	23	e2@op.pl
5	Jasio	34	NULL
6	Radek	54	e3@op.pl
7	Marek	23	NULL
8	Jan	33	e4@op.pl
9	Olek	55	e5@op.pl
10	Maria	22	NULL
11	Franek	44	NULL
12	Maja	44	NULL

ID	IMIE	WIEK	email
2	Marek	45	e1@op.pl
3	Marta	34	NULL
7	Marek	23	NULL
10	Maria	22	NULL
12	Maja	44	NULL



**UWAGA:** Zwróć uwagę, że tworząc warunki dotychczas odwoływaliśmy się do danych typu liczbowego (wiek, identyfikator...). Kiedy warunek tyczył się wartości typu liczba – np. WIEK, wartość warunku była podana bez żadnych cudzysłówów:

**WIEK > 20**

Jednak kiedy odwołujemy się w warunku do danych typu tekst, dane w warunku muszą być objęte pojedynczym cudzysłowem:

**IMIE = 'MAREK'**

**IMIE like 'Ma%'**

## LIKE raz jeszcze

```
SELECT * FROM users WHERE imie LIKE '_a%'
```

ID	IMIE	WIEK	email
1	Jacek	43	NULL
2	Marek	45	e1@op.pl
3	Marta	34	NULL
5	Jasio	34	NULL
6	Radek	54	e3@op.pl
7	Marek	45	NULL
8	Jan	33	e4@op.pl
10	Maria	22	NULL
12	Maja	44	NULL

Wypisane zostały te osoby, które na drugiej pozycji w imieniu mają literę 'A'

## Operator IN

Za pomocą tego operatora możemy sprawdzić czy wartość należy do danego zbioru.

Zamiast zapisywać zapytanie w ten sposób:

```
SELECT imie FROM users WHERE id =1 OR id=5 OR id=10;
```

Można te same wyniki uzyskać w ten sposób:

```
SELECT imie FROM users WHERE id IN (1,5,10);
```

Określiliśmy tu nic innego jak warunek, że chcemy uzyskać imiona tych userów, których identyfikator jest w zbiorze wartości wymienionych w nawiasie...

**SELECT imie FROM users WHERE id IN (1,5,10);**

ID	IMIE	WIEK	email
1	Jacek	43	NULL
2	Marek	45	e1@op.pl
3	Marta	34	NULL
4	Jola	23	e2@op.pl
5	Jasio	34	NULL
6	Radek	54	e3@op.pl
7	Marek	23	NULL
8	Jan	33	e4@op.pl
9	Olek	55	e5@op.pl
10	Maria	22	NULL
11	Franek	44	NULL
12	Maja	44	NULL



imie
Jacek
Jasio
Maria

**ĆWICZENIE: Wypisz dane kotów, których imię zawiera literę 'a'.**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
4	lady gada	5	3
5	myszka	2	2
7	marmelada	2	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
13	czarnulka	4	3
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

**SELECT \* FROM koty WHERE imie LIKE '%a%'**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
4	lady gada	5	3
5	myszka	2	2
7	marmelada	2	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
13	czarnulka	4	3
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3
20	stasiu	2	1

**ĆWICZENIE: Wypisz dane kotów, których wiek jest wyższy niż 2 lata i imię zawiera literę 'a'.**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
4	lady gada	5	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
13	czarnulka	4	3
17	franka	7	3
19	milka	3	3

**SELECT \* FROM koty WHERE wiek > 2 and imie LIKE '%a%'**

id_k	imie	wiek	id_bandy
1	gapcio	3	1
2	fruzia	12	1
4	lady gada	5	3
8	leopard	9	1
9	szemrak	12	1
10	macius	23	1
13	czarnulka	4	3
17	franka	7	3
19	milka	3	3



**ĆWICZENIE:** za pomocą operatora IN wyszukaj koty z band o numerach 2 i 3.

id_k	imie	wiek	id_bandy
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
7	marmelada	2	3
13	czarnulka	4	3
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3

**SELECT \* FROM koty WHERE id\_bandy IN (2,3)**

id_k	imie	wiek	id_bandy
3	smykos	4	2
4	lady gada	5	3
5	myszka	2	2
7	marmelada	2	3
13	czarnulka	4	3
16	zadziora	2	2
17	franka	7	3
18	malutka	2	2
19	milka	3	3

```
SELECT * FROM users ORDER BY imie, id DESC;
```

### **Klauzula ORDER BY**

Za pomocą tej klauzuli możemy określić kolumny według której będą posortowane wyniki. Domyślnie wyniki sortowane są od najmniejszego do największego (ASC), możemy to zmienić dodając po nazwie pola słowo DESC. Należy także wspomnieć, że można podać kilka kolumn według których dane będą sortowane, w przypadku gdy pierwsze wyrażenie będzie takie samo dla kilku pól to pod uwagę będzie brane drugie wyrażenie itd.

**SELECT \* FROM users ORDER BY imie, id DESC;**

ID	IMIE	WIEK	email
1	Jacek	43	NULL
2	Marek	45	e1@op.pl
3	Marta	34	NULL
4	Jola	23	e2@op.pl
5	Jasio	34	NULL
6	Radek	54	e3@op.pl
7	Marek	23	NULL
8	Jan	33	e4@op.pl
9	Olek	55	e5@op.pl
10	Maria	22	NULL
11	Franek	44	NULL
12	Maja	44	NULL




ID	IMIE	WIEK	email
11	Franek	44	NULL
1	Jacek	43	NULL
8	Jan	33	e4@op.pl
5	Jasio	34	NULL
4	Jola	23	e2@op.pl
12	Maja	44	NULL
7	Marek	23	NULL
2	Marek	45	e1@op.pl
10	Maria	22	NULL
3	Marta	34	NULL
9	Olek	55	e5@op.pl
6	Radek	54	e3@op.pl

**SELECT \* FROM users ORDER BY imie DESC, id ASC;**


ID	IMIE	WIEK	email
6	Radek	54	e3@op.pl
9	Olek	55	e5@op.pl
3	Marta	34	NULL
10	Maria	22	NULL
2	Marek	45	e1@op.pl
7	Marek	23	NULL
12	Maja	44	NULL
4	Jola	23	e2@op.pl
5	Jasio	34	NULL
8	Jan	33	e4@op.pl
1	Jacek	43	NULL
11	Franek	44	NULL

**ORDER BY imie, id DESC;**

**ORDER BY imie DESC, id ASC;**



12	Maja	44	NULL
7	Marek	23	NULL
2	Marek	45	e1@op.pl
10	Maria	22	NULL



10	Maria	22	NULL
2	Marek	45	e1@op.pl
7	Marek	23	NULL
12	Maja	44	NULL

## SELECT \* FROM users GROUP BY wiek;

Dzięki klauzuli GROUP BY możemy pogrupować wiersze według danego pola.

Powyższe zapytanie zwróci wyniki w postaci kolejnych wierszy pogrupowanych według wieku.

ID	IMIE	WIEK	email
10	Maria	22	NULL
7	Marek	23	NULL
4	Jola	23	e2@op.pl
8	Jan	33	e4@op.pl
3	Marta	34	NULL
5	Jasio	34	NULL
1	Jacek	43	NULL
11	Franek	44	NULL
12	Maja	44	NULL
2	Marek	45	e1@op.pl
6	Radek	54	e3@op.pl
9	Olek	55	e5@op.pl

**UWAGA:** klauzula ta nie zwraca powtarzających się wierszy.

**ĆWICZENIE: Wypisz dane kotów, posortowane malejąco względem numeru bandy.**

id_k	imie	wiek	id_bandy ▾ 1
13	czarnulka	4	3
17	franka	7	3
7	marmelada	2	3
4	lady gada	5	3
19	milka	3	3
5	myszka	2	2
18	malutka	2	2
3	smykos	4	2
16	zadziora	2	2
15	bury	7	1
14	zbyszko	2	1
1	gapcio	3	1
12	mruczus	6	1
11	zdzisio	9	1
10	macius	23	1
9	szemrak	12	1
8	leopard	9	1
6	gotfryd	7	1
2	fruzia	12	1
20	stasiu	2	1

**SELECT \* FROM koty ORDER BY id\_bandy DESC**

id_k	imie	wiek	id_bandy
13	czarnulka	4	3
17	franka	7	3
7	marmelada	2	3
4	lady gada	5	3
19	milka	3	3
5	myszka	2	2
18	malutka	2	2
3	smykos	4	2
16	zadziora	2	2
15	bury	7	1
14	zbyszko	2	1
1	gapcio	3	1
12	mruczus	6	1
11	zdzisio	9	1
10	macius	23	1
9	szemrak	12	1
8	leopard	9	1
6	gotfryd	7	1
2	fruzia	12	1
20	stasiu	2	1



**ĆWICZENIE: Wypisz dane kotów posortowane malejąco względem wieku i rosnąco względem numeru bandy.**

id_k	imie	wiek ▾ 1	id_bandy ▲ 2
10	macius	23	1
2	fruzia	12	1
9	szemrak	12	1
11	zdzisio	9	1
8	leopard	9	1
15	bury	7	1
6	gotfryd	7	1
17	franka	7	3
12	mruczus	6	1
4	lady gada	5	3
3	smykos	4	2
13	czarnulka	4	3
1	gapcio	3	1
19	milka	3	3
14	zbyszko	2	1
20	stasiu	2	1
16	zadziora	2	2
5	myszka	2	2
18	malutka	2	2
7	marmelada	2	3

**SELECT \* FROM koty ORDER BY wiek DESC, id\_bandy ASC**

id_k	imie	wiek ▾ 1	id_bandy ▲ 2
10	macius	23	1
2	fruzia	12	1
9	szemrak	12	1
11	zdzisio	9	1
8	leopard	9	1
15	bury	7	1
6	gotfryd	7	1
17	franka	7	3
12	mruczus	6	1
4	lady gada	5	3
3	smykos	4	2
13	czarnulka	4	3
1	gapcio	3	1
19	milka	3	3
14	zbyszko	2	1
20	stasiu	2	1
16	zadziora	2	2
5	myszka	2	2
18	malutka	2	2
7	marmelada	2	3

**ĆWICZENIE: Wypisz imiona oraz numery band kotów pogrupowane względem numeru bandy.**

imie	id_bandy
lady gada	3
smykos	2
gapcio	1

**SELECT imie, nr\_bandy FROM koty GROUP BY id\_bandy**

<b>imie</b>	<b>id_bandy</b> ▲ 2
lady gada	3
smykos	2
gapcio	1

**ĆWICZENIE: Wypisz dane kotów, które są starsze niż 5 lat posortowane względem wieku malejąco.**

id_k	imie	wiek	id_bandy
10	macius	23	1
2	fruzia	12	1
9	szemrak	12	1
8	leopard	9	1
11	zdzisio	9	1
6	gotfryd	7	1
15	bury	7	1
17	franka	7	3
12	mruczus	6	1

**SELECT \* FROM koty WHERE wiek > 5 ORDER BY wiek DESC**

id_k	imie	wiek	id_bandy
10	macius	23	1
2	fruzia	12	1
9	szemrak	12	1
8	leopard	9	1
11	zdzisio	9	1
6	gotfryd	7	1
15	bury	7	1
17	franka	7	3
12	mruczus	6	1

**ĆWICZENIE: Wyświetl dane kotów, których imię zawiera literę 'a' posortowane względem numeru bandy malejąco.**

id_k	imie	wiek	id_bandy ▾ 1
19	milka	3	3
4	lady gada	5	3
7	marmelada	2	3
17	franka	7	3
13	czarnulka	4	3
5	myszka	2	2
18	malutka	2	2
16	zadziora	2	2
1	gapcio	3	1
10	macius	23	1
9	szemrak	12	1
8	leopard	9	1
2	fruzia	12	1
20	stasiu	2	1

**SELECT \* FROM koty WHERE imie LIKE '%a%' ORDER BY id\_bandy DESC**

id_k	imie	wiek	id_bandy
19	milka	3	3
4	lady gada	5	3
7	marmelada	2	3
17	franka	7	3
13	czarnulka	4	3
5	myszka	2	2
18	malutka	2	2
16	zadziora	2	2
1	gapcio	3	1
10	macius	23	1
9	szemrak	12	1
8	leopard	9	1
2	fruzia	12	1
20	stasiu	2	1



## Funkcje agregujące

Za pomocą tych funkcji możemy uzyskać różne informacje statystyczne.

Dostępne funkcje agregujące to:

**COUNT(kolumna)** - zlicza liczbę pól występujących w kolumnie

**SUM(kolumna)** - sumuje wszystkie wartości w danej kolumnie

**MAX(kolumna)** - zwraca największą wartość w danej kolumnie

**MIN(kolumna)** - zwraca najmniejszą wartość w danej kolumnie

**AVG(kolumna)** - zwraca średnią wartość w danej kolumnie

**SELECT count(imie) FROM users WHERE id > 5;**

Przykład zwróci liczbę użytkowników (nie imiona tylko ich liczbę), którzy mają id większe od 5:

count(*)
7

**SELECT MAX(wiek) FROM users;**

Przykład zwróci wiek najstarszego usera:

MAX(wiek)
55

~~**SELECT MAX(wiek), imie FROM users;**~~

MAX(wiek)	imie
55	Jacek

~~Przykład dokładnie jak wyżej, ale poznamy również imie delikwenta:~~

Funkcje agregujące wyświetlają pewną wartość w danej kolumnie (największą, najmniejszą, średnią), ale wartość ta nie jest skorelowana z konkretnym wierszem do którego się odnosi.  
W tym wypadku dla kolumny wiek wycięgnięta została najmniejsza wartość WIEK dla kota. Imie Jacek nie oznacza, że to on posiada ten wiek, poprostu do minimalnego wieku kota baza 'dokleiła' imię pierwszego kota z listy kotów...!!!

# Klauzula LIMIT

Bardzo przydatna klauzula za pomocą której możemy określić ilość rekordów które chcemy pobrać.

**SELECT \* FROM users LIMIT 3;**

w ten sposób pobierzemy trzy rekordy.

ID	IMIE	WIEK	email
1	Jacek	43	NULL
2	Marek	45	e1@op.pl
3	Marta	34	NULL

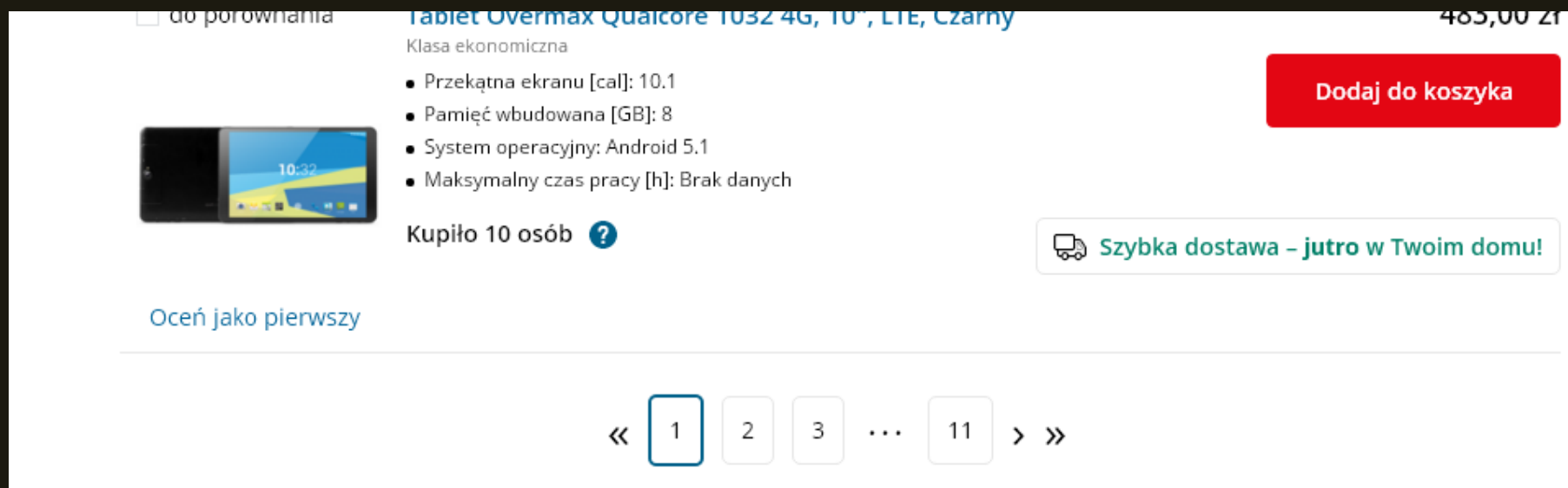
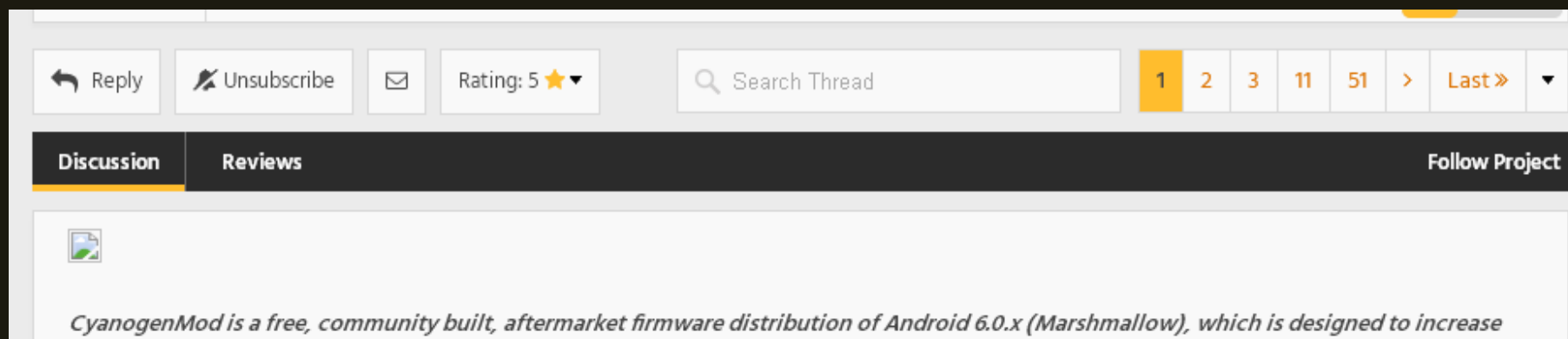
Istnieje też drugi sposób zapisu który często jest wykorzystywany m.in. przy stronicowaniu danych:

**SELECT \* FROM users LIMIT 3,5;**

Pobierze 5 rekordów licząc od trzeciego wiersza.

ID	IMIE	WIEK	email
4	Jola	23	e2@op.pl
5	Jasio	34	NULL
6	Radek	54	e3@op.pl
7	Marek	23	NULL
8	Jan	33	e4@op.pl

# Gdzie można się spotkać z powszechnym stosowaniem klauzuli LIMIT:



**ZŁĄCZENIA TABEL**

**Jak już wiemy, mało kiedy w bazie danych mamy do czynienia z jedną tabelą. Najczęściej tabel jest wiele i dodatkowo są między nimi związki (czyli dane z jednej tabeli są powiązane z danymi z innej tabeli).**

**W takim wypadku kwestią czasu będzie, aż znajdzie potrzeba wyszukania w bazie danych z kilku tabel naraz uwzględniając zachodzące między nimi powiązania...**

**...Wtedy należy stosować tak zwane złączenia tabel...**

## DANE

### PRACOWNIK

nazwisko	ID_dzialu
Kulikowski	103
Choromański	100
Szklanko	102
Nowik	103
Kowalski	101
Zabielski	NULL

### DZIAŁ

ID_dzialu	nazwa_dzialu
100	sprzedaz
101	inzynieria
102	marketing
103	administracja

# INNER JOIN

Jest to najbardziej popularna operacja używana w operacjach.

Efektem jest kombinacja wartości z 2 tabeli które mają wspólny element wspólny , wskazany w komendzie jako "łącznik" tabel.

**Elementy według których dokonywane są dopasowania nie muszą nazywać się identycznie, istotne natomiast są wartości aby były tego samego typu o były porównywalne.**

```
SELECT * FROM pracownik  
INNER JOIN dzial ON pracownik.ID_dzialu = dzial.ID_dzialu
```

<b>nazwisko</b>	<b>ID_dzialu</b>	<b>ID_dzialu</b>	<b>nazwa_dzialu</b>
Choromański	100	100	sprzedaz
Kowalski	101	101	inzynieria
Szklanko	102	102	marketing
Kulikowski	103	103	administracja
Nowik	103	103	administracja



Komendę możemy wzbogacić oczywiście o definicję warunków, sortowań, itd., np:

```
SELECT *FROM pracownik  
INNER JOIN dzial ON pracownik.ID_dzialu = dzial.ID_dzialu  
WHERE dzial.ID_dzialu<103 ORDER BY nazwisko DESC
```

<b>nazwisko</b> ▼	<b>ID_dzialu</b>	<b>ID_dzialu</b>	<b>nazwa_dzialu</b>
Szklanko	102	102	marketing
Kowalski	101	101	inzynieria
Choromański	100	100	sprzedaz

Można również zastosować bardziej wyraźny zapis, pozbawiony INNER JOINA a wykorzystujący do łączenia tabel warunek **WHERE**:

```
SELECT * FROM pracownik, dzial  
WHERE pracownik.ID_dzialu = dzial.ID_dzialu
```

nazwisko	ID_dzialu	ID_dzialu	nazwa_dzialu
Choromański	100	100	sprzedaz
Kowalski	101	101	inzynieria
Szklanko	102	102	marketing
Kulikowski	103	103	administracja
Nowik	103	103	administracja

## PRACOWNIK

nazwisko	ID_dzialu
Kulikowski	103

## DZIAŁ

ID_dzialu	nazwa_dzialu
100	spredaz

**Uwaga:** Jeśli w dwóch różnych tabelach są atrybuty (kolumny) o tej samej nazwie chcąc wskazać, o który atrybut nam chodzi stosujemy zapis:

**TABELA**.atrybut → **pracownik**.ID\_dzialu lub **dzial**.ID\_dzialu

## PRACOWNIK

nazwisko	ID_dzialu
Kulikowski	103

## DZIAŁ

ID_dzialu	nazwa_dzialu
100	sprzedaz

Przykład:

```
SELECT nazwisko, nazwa_dzialu, dzial.ID_dzialu
```

```
FROM pracownik, dzial
```

```
WHERE pracownik.ID_dzialu = dzial.ID_dzialu;
```

Chcąc wypisać identyfikator działu (ID\_dzialu) należało wyspecyfikować czy ma to być atrybut z tabeli PRACOWNIK czy DZIAL (w obu jest taki atrybut).

Podobnie przy warunku...

# Natural join

Oferuje dalszą specjalizację equi-join.

"Łącznik" tabel pojawia się automatycznie poprzez porównywanie kolumn obu tabel na podstawie nazw kolumn które muszą być identyczne. np:

**SELECT \* FROM pracownik NATURAL JOIN dzial**

ID_dzialu	nazwisko	nazwa_dzialu
100	Choromański	sprzedaz
101	Kowalski	inzynieria
102	Szklanko	marketing
103	Kulikowski	administracja
103	Nowik	administracja

Celem wyszukania wszystkich rekordów z jednej tabeli i tylko pasujących z drugiej używamy LEFT JOIN oraz RIGHT JOIN, polecenie left i right wskazuje która tabela ma być wyświetlana w całości.

```
SELECT *FROM pracownik  
LEFT JOIN dzial ON pracownik.ID_dzialu = dzial.ID_dzialu'
```

nazwisko	ID_dzialu	nazwa_dzialu
Choromański	100	sprzedaz
Kowalski	101	inzynieria
Szklanko	102	marketing
Kulikowski	103	administracja
Nowik	103	administracja

**PLUS  
DODATKOWO POJAWI SIĘ NA LIŚCIE OSOBA**

Zabielski	NULL	NULL
-----------	------	------

**ĆWICZENIE: Wyszukaj dane kotów oraz ich band , stosując mechanizm inner join.**

id_k	imie	wiek	id_bandy	id_b	nazwa	id_szefa
1	gapcio	3	1	1	dachowce	3
2	fruzia	12	1	1	dachowce	3
6	gotfryd	7	1	1	dachowce	3
8	leopard	9	1	1	dachowce	3
9	szemrak	12	1	1	dachowce	3
10	macius	23	1	1	dachowce	3
11	zdzisio	9	1	1	dachowce	3
12	mruczus	6	1	1	dachowce	3
14	zbyszko	2	1	1	dachowce	3
15	bury	7	1	1	dachowce	3
20	stasiu	2	1	1	dachowce	3
3	smykos	4	2	2	rakiety	19
5	myszka	2	2	2	rakiety	19
16	zadziora	2	2	2	rakiety	19
18	malutka	2	2	2	rakiety	19
4	lady gada	5	3	3	szalone samice	14
7	marmelada	2	3	3	szalone samice	14
13	czarnulka	4	3	3	szalone samice	14
17	franka	7	3	3	szalone samice	14
19	milka	3	3	3	szalone samice	14

**SELECT \* FROM koty INNER JOIN bandy ON id\_bandy = id\_b**

id_k	imie	wiek	id_bandy	id_b	nazwa	id_szefa
1	gapcio	3	1	1	dachowce	3
2	fruzia	12	1	1	dachowce	3
6	gotfryd	7	1	1	dachowce	3
8	leopard	9	1	1	dachowce	3
9	szemrak	12	1	1	dachowce	3
10	macius	23	1	1	dachowce	3
11	zdzisio	9	1	1	dachowce	3
12	mruczus	6	1	1	dachowce	3
14	zbyszko	2	1	1	dachowce	3
15	bury	7	1	1	dachowce	3
20	stasiu	2	1	1	dachowce	3
3	smykos	4	2	2	rakiety	19
5	myszka	2	2	2	rakiety	19
16	zadziora	2	2	2	rakiety	19
18	malutka	2	2	2	rakiety	19
4	lady gada	5	3	3	szalone samice	14
7	marmelada	2	3	3	szalone samice	14
13	czarnulka	4	3	3	szalone samice	14
17	franka	7	3	3	szalone samice	14
19	milka	3	3	3	szalone samice	14



**ĆWICZENIE: Wyszukaj dane kotów oraz ich band, stosując mechanizm łączenia tabel definiowany w warunku WHERE.**

id_k	imie	wiek	id_bandy	id_b	nazwa	id_szefa
1	gapcio	3	1	1	dachowce	3
2	fruzia	12	1	1	dachowce	3
6	gotfryd	7	1	1	dachowce	3
8	leopard	9	1	1	dachowce	3
9	szemrak	12	1	1	dachowce	3
10	macius	23	1	1	dachowce	3
11	zdzisio	9	1	1	dachowce	3
12	mruczus	6	1	1	dachowce	3
14	zbyszko	2	1	1	dachowce	3
15	bury	7	1	1	dachowce	3
20	stasiu	2	1	1	dachowce	3
3	smykos	4	2	2	rakiety	19
5	myszka	2	2	2	rakiety	19
16	zadziora	2	2	2	rakiety	19
18	malutka	2	2	2	rakiety	19
4	lady gada	5	3	3	szalone samice	14
7	marmelada	2	3	3	szalone samice	14
13	czarnulka	4	3	3	szalone samice	14
17	franka	7	3	3	szalone samice	14
19	milka	3	3	3	szalone samice	14

**SELECT \* FROM koty, bandy WHERE id\_bandy = id\_b**

id_k	imie	wiek	id_bandy	id_b	nazwa	id_szefa
1	gapcio	3	1	1	dachowce	3
2	fruzia	12	1	1	dachowce	3
6	gotfryd	7	1	1	dachowce	3
8	leopard	9	1	1	dachowce	3
9	szemrak	12	1	1	dachowce	3
10	macius	23	1	1	dachowce	3
11	zdzisio	9	1	1	dachowce	3
12	mruczus	6	1	1	dachowce	3
14	zbyszko	2	1	1	dachowce	3
15	bury	7	1	1	dachowce	3
20	stasiu	2	1	1	dachowce	3
3	smykos	4	2	2	rakiety	19
5	myszka	2	2	2	rakiety	19
16	zadziora	2	2	2	rakiety	19
18	malutka	2	2	2	rakiety	19
4	lady gada	5	3	3	szalone samice	14
7	marmelada	2	3	3	szalone samice	14
13	czarnulka	4	3	3	szalone samice	14
17	franka	7	3	3	szalone samice	14
19	milka	3	3	3	szalone samice	14

**ĆWICZENIE:** Wyszukaj dane kotów (id\_k, imie, wiek) oraz ich band (id\_b, nazwa), stosując mechanizm łączenia tabel definiowany w warunku WHERE oraz posortowanych względem wieku kotów malejąco.

id_k	imie	wiek	id_b	nazwa
10	macius	23	1	dachowce
2	fruzia	12	1	dachowce
9	szemrak	12	1	dachowce
8	leopard	9	1	dachowce
11	zdzisio	9	1	dachowce
15	bury	7	1	dachowce
6	gotfryd	7	1	dachowce
17	franka	7	3	szalone samice
12	mruczus	6	1	dachowce
4	lady gada	5	3	szalone samice
3	smykos	4	2	rakiety
13	czarnulka	4	3	szalone samice
19	milka	3	3	szalone samice
1	gapcio	3	1	dachowce
18	malutka	2	2	rakiety
5	myszka	2	2	rakiety
20	stasiu	2	1	dachowce
16	zadziora	2	2	rakiety
14	zbyszko	2	1	dachowce
7	marmelada	2	3	szalone samice

```
SELECT id_k, imie, wiek, id_b, nazwa FROM koty, bandy
```

```
WHERE id_bandy = id_b ORDER BY wiek DESC
```

id_k	imie	wiek	id_b	nazwa
10	macius	23	1	dachowce
2	fruzia	12	1	dachowce
9	szemrak	12	1	dachowce
8	leopard	9	1	dachowce
11	zdzisio	9	1	dachowce
15	bury	7	1	dachowce
6	gotfryd	7	1	dachowce
17	franka	7	3	szalone samice
12	mruczus	6	1	dachowce
4	lady gada	5	3	szalone samice
3	smykos	4	2	rakiety
13	czarnulka	4	3	szalone samice
19	milka	3	3	szalone samice
1	gapcio	3	1	dachowce
18	malutka	2	2	rakiety
5	myszka	2	2	rakiety
20	stasiu	2	1	dachowce
16	zadziora	2	2	rakiety
14	zbyzsko	2	1	dachowce
7	marmelada	2	3	szalone samice

**ĆWICZENIE:** Wyszukaj dane kotów (id\_k, imie, wiek) oraz ich band (id\_b, nazwa), ale uwzględniając tylko te koty, które są starsze niż 5 lat.

id_k	imie	wiek	id_b	nazwa
2	fruzia	12	1	dachowce
6	gotfryd	7	1	dachowce
8	leopard	9	1	dachowce
9	szemrak	12	1	dachowce
10	macius	23	1	dachowce
11	zdzisio	9	1	dachowce
12	mruczus	6	1	dachowce
15	bury	7	1	dachowce
17	franka	7	3	szalone samice

**SELECT id\_k, imie, wiek, id\_b, nazwa FROM koty, bandy**

**WHERE id\_bandy = id\_b and wiek > 5**

<b>id_k</b>	<b>imie</b>	<b>wiek</b>	<b>id_b</b>	<b>nazwa</b>
2	fruzia	12	1	dachowce
6	gotfryd	7	1	dachowce
8	leopard	9	1	dachowce
9	szemrak	12	1	dachowce
10	macius	23	1	dachowce
11	zdzisio	9	1	dachowce
12	mruczus	6	1	dachowce
15	bury	7	1	dachowce
17	franka	7	3	szalone samice

**ĆWICZENIE: Wyszukaj dane band (id\_b, nazwa) wraz z imionami i wiekiem ich szefów.**

id_b	nazwa	imie	wiek
1	dachowce	smykos	4
2	rakiety	milka	3
3	szalone samice	zbyszko	2

**ĆWICZENIE: Wypisz dane zdobyczy (nazwa, wartość) wraz z nazwą bandy, która jest ich właścicielem.**

nazwa	wartosc	nazwa
puszka szprotek	6	dachowce
sztuczna mysz	34	dachowce
stare skarpety	1	dachowce
kapsel	1	dachowce
buda	78	dachowce
wedzona makrela	12	rakiety
ser	13	rakiety
zapas mleka	23	rakiety
paletko	54	rakiety
pilka	65	rakiety
ryby	45	rakiety
obrazek	11	rakiety
medalik	22	szalone samice
bawik	34	szalone samice
szyneczka	89	szalone samice

**SELECT id\_b, nazwa, imie, wiek FROM bandy, koty WHERE id\_szefa = id\_k**

id_b	nazwa	imie	wiek
1	dachowce	smykos	4
2	rakiety	milka	3
3	szalone samice	zbyszko	2

**SELECT zdobycze.nazwa, wartosc, bandy.nazwa FROM zdobycze, bandy**

**WHERE wlasciciel = id\_b**

nazwa	wartosc	nazwa
puszka szprotek	6	dachowce
sztuczna mysz	34	dachowce
stare skarpety	1	dachowce
kapsel	1	dachowce
buda	78	dachowce
wedzona makrela	12	rakiety
ser	13	rakiety
zapas mleka	23	rakiety
paletko	54	rakiety
pilka	65	rakiety
ryby	45	rakiety
obrazek	11	rakiety
medalik	22	szalone samice
bawik	34	szalone samice
szyneczka	89	szalone samice



**Koniec cz.1 ....**

**...C.D.N.**

**W dalszej części PODZAPYTANIA  
ZAGNIEŹDŻONE**