

Zajęcia 1

Sprawdzanie dostępnych baz danych:

```
SHOW DATABASES;
```

Uzyskiwanie pomocy:

```
HELP;
```

Pomoc dla metody:

```
HELP nazwa_metody;
```

Korzystanie z bazy, czyli ustawianie jej jako aktywnej:

```
USE nazwa_bazy;
```

Sprawdzanie jaka baza danych jest aktywna:

```
SELECT database();
```

Wyświetlanie listy tabel dostępnych w aktywnej bazie:

```
SHOW tables;
```

Wywoływanie opisu tabeli:

```
DESC nazwa_tabeli;
```

Tworzenie nowej bazy danych:

```
CREATE DATABASE nazwa_bazy;
```

Usuwanie bazy i zawartych w niej tabel:

```
DROP DATABASE nazwa_bazy;
```

Zadania

Utwórz bazę danych o nazwie zwierzęta, następnie ją usuń.

Wylistuj wszystkie bazy danych.

Typy danych

Interesować nas one będą przy tworzeniu tabel (jest ich oczywiście więcej, ale nas interesować będą na razie te poniższe).

Tekstowe:

var 0-255 np. char(10) varchar 0 – 65535 varchar(344)

liczbowe

float – nie gubi dokładności (7 cyfr po przecinku) , double – podwójna precyzja (15 cyfr, 8 bitów)

Tworzenie tabeli

```
CREATE TABLE nazwa_tabeli  
  
(  
  
nazwa_kolumny typ_danych,  
  
nazwa_kolumny typ_danych  
  
);
```

```
DESC nazwa_tabeli;
```

lub:

```
SHOW COLUMNS FROM nazwa_tabeli;
```

Dodawanie do niej wartości

```
INSERT INTO nazwa tabeli (kolumny) VALUES (wartości); nazwy kolumn nie trzeba  
podawać w kolejności występującej w tabeli, ale później, wartości należy wstawić w takiej  
kolejności jak podano w zapytaniu
```

***Jeżeli nie chcemy żeby pole przy wyborze tabeli było puste musimy dodać po typie danych NOT NULL).**

Pominięcie podania wartości dla kolumny oznaczonej jako NOT NULL spowoduje wyświetlenie błędu.

Możemy też dać wartość w przypadku jeśli nie poda użytkownik nic w polu, które zdefiniujemy jako NOT NULL.

Wygląda to tak:

```
nazwa CHAR(20) NOT NULL DEFAULT "nazwa która ma się pojawić po niewpisaniu  
przez użytkownika",
```

```
SELECT * FROM nazwa_tabeli;
```

Wyświetlanie określonych kolumn z tabeli:

```
SELECT nazwa_kolumny, nazwa_kolumny2 FROM nazwa_tabeli;
```

Usuwanie tabeli

```
DROP TABLE nazwa_tabeli;
```

Usuwanie tabeli z bazy danych innej niż aktywna:

```
DROP TABLE nazwa_bazy.nazwa_tabeli;
```

Zadania

Utwórz tabelę zwierzęta zawierającą trzy kolumny: nazwa, waga, gatunek.

Nazwa i gatunek mają być typem znakowym (nazwa - 20 znaków, gatunek - 10 znaków, waga typ całkowity).

Wstaw do tabeli dla przykładu 10 rekordów (rekord to jeden wiersz w tabeli).

Wstawiamy w następujący sposób:

```
INSERT INTO nazwa_tabeli (nazwy kolumn) VALUES (wartości które chcemy wprowadzić)
```

Np:

```
INSERT INTO zwierzęta VALUES ('Kot', 5, 'ssak');
```

Jeżeli chcemy od razu wprowadzić więcej wartości to zamiast średnika dajemy przecinek i podajemy kolejne wartości w nawiasie.

1. Kot, 5, ssak
2. Pies, 10, ssak
3. Żółw, 10, gad
4. Wieloryb, 2000, ssak
5. Czapla, 15, ptak
6. Koliber, 1, ptak
7. Wąż, 10, gad
8. Jaszczurka, 2, gad
9. Lew, 150, ssak
10. Hipopotam, 800, ssak

Fraza LIMIT służy do ograniczania wyniku zapytania, na przykład po pierwszych 5 przypadków:

```
SELECT length, title FROM film LIMIT 5;
```

Fraza ORDER BY – porządkowanie danych (rosnące lub malejące – domyślnie jest rosnące: ASC, malejące: DESC).

```
SELECT * FROM zwierzęta ORDER BY (po jakiej kolumnie sortujemy, np. nazwa) nazwa  
(teraz podajemy czy ma być malejąco) DESC
```

Fraza WHERE – gdzie.

```
SELECT * (gwiazdka oznacza wszystkie rekordy) FROM nazwa_tabeli
```

```
WHERE jakiś warunek np. waga >=300
```

Jeżeli chcemy mieć większe równe 300 i mniejsze od 800 możemy zrobić to z operatorem AND lub OR.

```
WHERE waga >=300 AND waga <800
```

Zamiast dwóch warunków możemy użyć BETWEEN:

```
WHERE waga BETWEEN 300 AND 799
```

DISTINCT – eliminuje powtarzające się wiersze

Porównując wartości w MySQL możemy stosować operatory:

= równości

< jest mniejszy niż

> jest większy niż

<= jest mniejszy lub równy

>= jest większy lub równy

!= jest różny od

przykładowo jeżeli chcemy znaleźć zwierzęta których waga wynosi dokładnie 130 napiszemy:

```
SELECT * FROM zwierzęta WHERE waga = 130;
```

jeżeli natomiast chcemy wybrać wszystko poza tą wagą, napiszemy:

```
SELECT * FROM zwierzęta WHERE waga != 130;
```

Danej kolumny nie musimy wybierać aby obliczyć z niej warunek. Wyniki możemy też ograniczać słowem LIMIT:

```
SELECT nazwa FROM zwierzęta WHERE waga = 130 LIMIT 3;
```

Warunki związane z tekstem:

wykorzystujemy tutaj konstrukcję LIKE lub NOT LIKE

- % oznacza wszystkie znaki

- ? oznacza pojedynczy znak

Np.:

wyszukaj w bazie wszystkie zwierzęta na literkę s

```
SELECT nazwa FROM zwierzęta WHERE nazwa LIKE 's%';
```

NOT LIKE spowoduje wylistowanie wszystkich nie na literkę 's'.

Zadania

Wyświetl 5 pierwszych zwierząt podając ich nazwę i gatunek.

Wyświetl nazwy najcięższych 5 zwierząt oraz najlżejszych.

Wyświetl nazwy zwierząt i gatunek gdzie w nazwie o oraz waga jest większa niż 50.

Wyświetl najlżejsze zwierzę.