

### Nawiązanie połączenia z serwerem z konsoli systemu WINDOWS

Uruchamiamy serwer przez XAMPP Control Panel (moduły Apache i MySQL)

Z poziomu wiersza polecenia wybieramy lokalizację folderu mysql/bin

np.: **cd c:/xampp/mysql/bin**

Logowanie do serwera:

**mysql -u root [-p jeśli jest hasło]**

Kończenie połączenia:

**exit lub quit**

### UWAGA!

Instrukcje SQL można zapisać w pliku tekstowym (skrypcie) i wczytać je odpowiednim poleceniem.

W pliku możemy umieszczać komentarze (początek komentarza **/** a koniec komentarza **\*/**)

Uruchamianie skryptu (plik.txt lub plik.sql) zawierającego instrukcje napisane w SQL.

**SOURCE C:/Folder/plik.txt;**

lub **SOURCE C:/Folder/plik.sql;**

Wyświetlenie wszystkich dostępnych po zalogowaniu baz danych:

**SHOW DATABASES;**

Wyświetlenie, jaka baza danych jest obecnie używana:

**SELECT DATABASE();**

Jeśli nie wybrano żadnej bazy pojawi się **NULL**

Wyświetlanie bieżącego użytkownika:

**SELECT USER();**

Wybór bazy danych:

**USE nazwa\_bazy;**

Tworzenie bazy danych:

**CREATE DATABASE kowalski ;**

Jeśli baza o tej nazwie już istnieje pojawi się błąd. Można użyć konstrukcji warunkowej:

**CREATE DATABASE IF NOT EXISTS kowalski;**

Usuwanie bazy danych

Utworzenie

**CREATE DATABASE dousuwania;**

Usunięcie

**DROP DATABASE dousuwania;**

Jeśli nie wiemy czy baza istnieje można użyć konstrukcji:

**DROP DATABASE IF EXISTS dousuwania;**

Wybór nowej bieżącej bazy danych:

**USE kowalski;**

## **TABELE**

Wyświetlenie tabel bieżącej bazy danych:

**SHOW TABLES;**

Tworzenie tabeli w bieżącej bazie danych (IF NOT EXISTS):

```
CREATE TABLE osoby  
(  
  id INT(11) PRIMARY KEY auto_increment,  
  nazwisko VARCHAR(20),  
  imie VARCHAR(20),  
  miasto VARCHAR(20)  
);
```

Sprawdzanie struktury tabeli:

**DESCRIBE osoby;**

lub

**DESC osoby;**

Wyświetlanie zawartości tabeli

**SELECT \* FROM osoby;**

Dodawanie nowych rekordów

```
INSERT INTO osoby (ID, imie, nazwisko) VALUES (1, 'Kowalski', 'Jan');  
INSERT INTO osoby VALUES (2, 'Nowak', 'Anna', 'Cekcyn');
```

Wprowadzanie wielu wartości do wybranych kolumn

```
INSERT INTO osoby (nazwisko, imie, miasto) VALUES  
('Kotek', 'Jerzy', 'Tuchola'),  
('Kosowska', 'Maria', 'Chojnice'),  
('Zielinska', 'Zofia', 'Gostycyn'),  
('Marona', 'Filip', 'Tuchola');
```

Usuwanie tabeli

Utworzenie tabeli do usunięcia:

```
CREATE TABLE dousuwania  
(  
  id INT(11) PRIMARY KEY auto_increment,  
  haslo VARCHAR(20)  
);
```

Sprawdzenie **SHOW TABLES;**

Usunięcie tabeli (IF EXISTS):

**DROP TABLE dousuwania;**

lub

**DROP TABLE IF EXISTS dousuwania;**

Tworzenie tabeli w oparciu o inną tabelę (inne tabele): **CREATE TABLE ... AS SELECT ....**

Ciekawą możliwością jest tworzenie tabel w oparciu o inne, istniejące już tabele. Możliwość taka przydaje się wówczas, gdy np. w szybki sposób chcemy utworzyć kopię istniejącej tabeli (gdy np. planujemy eksperymentowanie z pewną tabelą i chcemy się zabezpieczyć przed omyłkowym uszkodzeniem danych).

**PRZYKŁAD**

```
CREATE TABLE kopia_osoby AS SELECT * FROM osoby;
SHOW TABLES;
SELECT * FROM kopia_osoby;
DROP TABLE kopia_osoby;
SHOW TABLES;
```

Dane można wczytać do istniejącej tabeli z pliku tekstowego poleceniem:

**LOAD DATA LOCAL INFILE "C:/.../nazwa\_pliku.txt" INTO TABLE nazwa\_tabeli;**

Domyślnie instrukcja Load Data zakłada, że wartości kolumn będą ograniczone przez tabulatory i że wiersze będą się kończyć przy użyciu znaków końca wierszy. Zakłada się również, że wartości występują w kolejności, w której kolumny są przechowywane w tabeli. Możliwe jest zdefiniowanie innego formatu pliku lub określenie innego porządku kolumn.

**LOAD DATA [LOW\_PRIORITY] [LOCAL] INFILE 'nazwa\_pliku.txt'**

Jeżeli podano słowo kluczowe LOCAL, plik jest odczytywany z hosta klienta.

**ZMIANA STRUKTURY TABELI (ALTER TABLE)**

Do zmiany struktury tabeli służy polecenie o następującej składni:

**ALTER TABLE nazwa\_tabeli zmiana1, zmiana2,...**

Instrukcja ALTER TABLE pozwala na zmianę struktury istniejącej tabeli: można dodawać lub usuwać kolumny, tworzyć lub usuwać indeksy, zmieniać typy lub kolejność istniejących kolumn oraz zmieniać nazwę kolumn lub tabel.

**ALTER TABLE nazwa\_tabeli parametry [, ...]**

**[ORDER BY kolumna]**

parametry:

```
ADD [COLUMN] definicja_kolumny [FIRST | AFTER nazwa_kolumny ]
lub ADD [COLUMN] (definicja_kolumny, definicja_kolumny,...)
lub ADD INDEX [nazwa_indeksu] (nazwa_kolumny_indeksowanej,...)
lub ADD PRIMARY KEY (nazwa_kolumny_indeksowanej,...)
lub ADD UNIQUE [nazwa_indeksu] (nazwa_kolumny_indeksowanej,...)
lub ADD FULLTEXT [nazwa_indeksu] (nazwa_kolumny_indeksowanej,...)
lub ADD [CONSTRAINT symbol] FOREIGN KEY nazwa_indeksu
    (nazwa_kolumny_indeksowanej,...)
    [definicja_referencji]
lub ALTER [COLUMN] nazwa_kolumny {SET DEFAULT literal | DROP DEFAULT}
lub CHANGE [COLUMN] stara_nazwa_kolumny definicja_kolumny
lub MODIFY [COLUMN] definicja_kolumny
lub DROP [COLUMN] nazwa_kolumny
lub DROP PRIMARY KEY
lub DROP INDEX nazwa_indeksu
lub RENAME [TO] nowa_nazwa_tabeli
lub opcje_tabeli
```

**PRZYKŁADY** (Po wykonaniu polecenia sprawdzaj zmiany w strukturze poleceniem DESCRIBE)

Zmiana typu danych w kolumnie

```
ALTER TABLE osoby  
MODIFY miasto VARCHAR(15);
```

Dodawanie kolumny w tabeli

```
ALTER TABLE osoby  
ADD plec ENUM('k', 'm') AFTER imie;
```

Modyfikacja danych w tabeli - dodawanie płci w tabeli osoby

```
UPDATE osoby SET plec='k' WHERE imie LIKE '%a';  
UPDATE osoby SET plec='m' WHERE imie NOT LIKE '%a';
```

Usuwanie rekordów z tabeli

```
DELETE FROM osoby WHERE imie LIKE 'J%';
```

Usuwanie kolumny

Dodanie kolumny dousuwania

```
ALTER TABLE osoby ADD dousuwania INT(5) FIRST;
```

Usunięcie kolumny dousuwania

```
ALTER TABLE osoby DROP dousuwania;
```

Zmiana nazwy tabeli

```
ALTER TABLE osoby RENAME TO osoby2;
```

Przywrócenie poprzedniej nazwy

```
ALTER TABLE osoby2 RENAME TO osoby;
```

Tworzenie indeksu

```
CREATE INDEX miasto_index ON osoby(miasto);
```

Wyświetlanie indeksów

```
SHOW INDEX FROM osoby;
```

Usuwanie indeksu

```
DROP INDEX miasto_index ON osoby;
```