

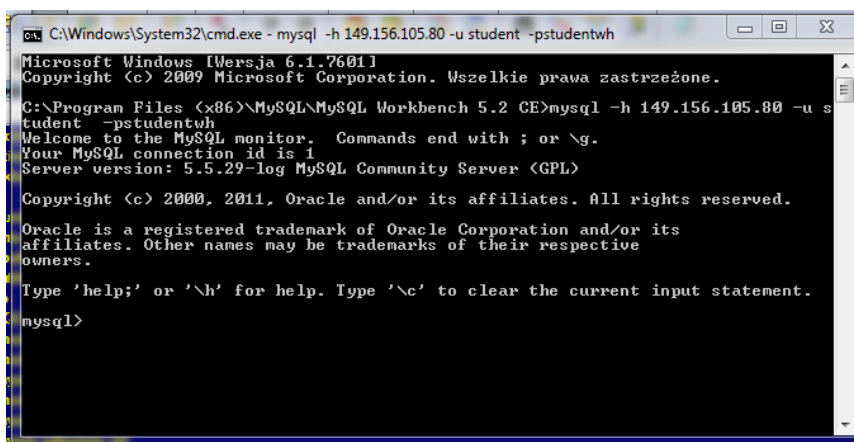
Ćwiczenia laboratoryjne nr 11 – Bazy danych i SQL.

Celem ćwiczeń jest poznanie zasad tworzenia baz danych i zastosowania komend SQL.

Ćwiczenie I. Logowanie do bazy danych.

Polecenie	Składnia ¹	Przykład
Logowanie	<code>mysql [-h localhost] [-u student] [-p]</code>	<ul style="list-style-type: none"><code>mysql -u student -pstudentwh</code>

- Przejdź do katalogu **C:\Program Files\krypt.sql\MySQL\MySQL Workbench 5.2 CE.**(lub innego w którym znajduje się MySQL Workbench.)
- Uruchom w nim procesor poleceń cmd.exe.
- Sprawdź, czy jest dostępny program mysql.exe.
- Jeśli tak, wpisz polecenie łączące z bazą danych na zdalnym komputerze.
 - mysql -h 149.156.105.80 -u student -pstudentwh**
 - W wyniku wykonania polecenia powinno się pojawić następujące okienko:**



```
ca: C:\Windows\System32\cmd.exe - mysql -h 149.156.105.80 -u student -pstudentwh
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Program Files (x86)\MySQL\MySQL Workbench 5.2 CE>mysql -h 149.156.105.80 -u student -pstudentwh
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.29-log MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

- W komendach nie używany znaków diaktrycznych (ś,ć,ż,ź,ą,ę,ń,...).
- W terminalu, po znaku zachęty, proszę wpisać następującą komendę:

tee d:\Imię_NazwiskoOut.txt

gdzie **Imię_Nazwisko**, to imię i nazwisko osoby wykonującej ćwiczenie. Polecenie powoduje, że wszystkie wykonywane polecenia są logowane do pliku **c:\Imię_NazwiskoOut.txt**.

¹ W opisie składni zastosowano: „[]” – element opcjonalny, „{}” – element obowiązkowy, „|” – alternatywa.

7. W oknie wpisz komendę:

ls

w celu uzyskania informacji statusie połączenia z bazą danych.

8. Należy sprawdzić, czy prawidłowo utworzył się plik logu *c:\Imię_NazwiskoOut.txt* i czy zawiera informację o statusie.

UWAGA !

Plik będzie podstawą do otrzymania zaliczenia z ćwiczeń. W związku z tym należy uzyskać pewność, że plik istnieje i dane są do niego zapisywane.

Ćwiczenie II. Wypisanie bazy danych istniejących na serwerze.

1. W okienku, po znaku zachęty, wpisz komendę:

SHOW DATABASES;

(pamiętaj o średniku: ;)

w celu wylistowania baz danych istniejących na serwerze.

UWAGA! Czasami połączenie jest przerywane (w celu optymalizacji obsługi bazy). Wtedy w linii poleceń należy wpisać polecenie „connect”.

Ćwiczenie III. Utworzenie bazy danych na serwerze.

1. W okienku, po znaku zachęty, wpisz następującą komendę:

CREATE DATABASE *Mojelnicy*Baza;

(pamiętaj o średniku: ;),

w celu utworzenia bazy danych o nazwie *Mojelnicy*baza (*Mojelnicy* są inicjałami osoby wykonującej ćwiczenie. Ma zapewnić unikalność nazwy bazy np. „**CREATE DATABASE wchbaza;**”).

2. Wpisz powtórnie komendę

SHOW DATABASES;

aby sprawdzić, czy baza została utworzona.

Ćwiczenie IV. Wybór bazy danych i utworzenie tabeli z danymi *pracownicy*.

1. W okienku, po znaku zachęty, wpisz następującą komendę:

USE *Mojelnicy*Baza;

w celu poinformowania serwera jaką bazę będziemy używać.

2. W okienku, po znaku zachęty, wpisz następującą komendę:

SHOW TABLES;

w celu sprawdzenia jakie tabele znajdują się w bazie. W tym momencie nie powinny istnieć żadne tabele w bazie.

3. W celu utworzenia tabeli **pracownicy**, zawierającą informację o pracownikach (**id, imie, nazwisko, miasto, staz, stanowisko, pensja**) wpisz następującą komendę:

```
CREATE TABLE pracownicy ( id INT, imie VARCHAR(20), nazwisko VARCHAR(20),  
miasto VARCHAR(20), staz INT, stanowisko VARCHAR(20), pensja DECIMAL(9,2));
```

a następnie sprawdź czy tabela istnieje, komendą **SHOW TABLES**;

Ćwiczenie V. Wypełnienie tabeli *pracownicy* danymi.

1. Wpisz następującą komendę:

```
INSERT INTO pracownicy values ('1', 'Roman', 'Zielinski', 'Warszawa', '2', 'Prawnik',  
'5000');
```

w celu wypełnienia danymi tabeli (1. wiersza).

2. W celu sprawdzenia zawartości tabeli wpisz komendę:

```
SELECT * FROM pracownicy;
```

Zwróć uwagę na to co zostało wypisane w terminalu.

3. W celu załadowania danych z pliku **pracownicy.txt**, po jego pobraniu (ze strony), wpisz komendę (zmień ścieżkę na rzeczywistą):

```
LOAD DATA LOCAL INFILE 'd:/pracownicy.txt' INTO TABLE pracownicy FIELDS  
TERMINATED BY ',';
```

4. Wypisz zawartość tabeli za pomocą następujących komend:

- **SELECT * FROM pracownicy;**
- **SELECT id FROM pracownicy;**
- **SELECT id ,imie, nazwisko FROM pracownicy;**
- **SELECT id , pensja FROM pracownicy;**

5. Wykonaj operację złożoną:

- **SELECT stanowisko, AVG(pensja) FROM pracownicy GROUP BY stanowisko;**

Ćwiczenie VI. Aktualizacja rekordów w tabeli *pracownicy*.

1. W terminalu wpisz następującą komendę:

```
UPDATE pracownicy SET pensja =6800 WHERE imie = 'Jan' AND nazwisko = 'Nowak';
```

w celu zmiany wysokości pensji pracownika.

2. Za pomocą komendy **SELECT** wypisz zawartość rekordu tylko dla Jana Nowaka:

```
SELECT * FROM pracownicy WHERE imie = 'Jan' AND nazwisko = 'Nowak';
```

Ćwiczenie VII. Wypełnienie tabeli *pracownicy* poprzez wykonanie skryptu SQL:

1. W celu załadowania danych z pliku *skrypt.sql*, po jego pobraniu (ze strony), wpisz komendę (zmień ścieżkę na rzeczywistą):

SOURCE d:/skrypt.sql;

2. Wypisz zawartość tabeli za pomocą następujących komend:

SELECT * FROM pracownicy;

Ćwiczenie VIII. Usuwanie rekordów z tabeli *pracownicy*:

1. W terminalu wpisz następującą komendę:

DELETE FROM pracownicy WHERE pensja = '6800' OR pensja = '6000';

w celu zmiany wysokości pensji pracownika.

2. Za pomocą komendy SELECT wypisz tablicę:

SELECT * FROM pracownicy;

Ćwiczenie IX. Wykonaj samodzielnie następujące operacje na bazie:

1. Dodaj rekord ze swoim *imieniem* i *nazwiskiem* (pozostałe pola wypełnij dowolnie wg uznania). W tym celu przepisz wartość polom rekordu takim jak: *Id, imie, nazwisko, miasto, staz, stanowisko, pensja* stosując odpowiednie zapytanie SQL (jedno z podanych wcześniej).
2. W celu sprawdzenia zawartości bazy wypisz zawartość wszystkich rekordów tabeli.
3. Wypisz całą zawartość tylko tych rekordów, gdzie *stanowisko = 'Asystent'*.
4. Zmień pensje wszystkim asystentom na 3000.
5. Wypisz tylko pensje dla stanowiska *Asystent*.
6. Usuń rekord, dla którego *nazwisko='Zalas'*. Sprawdź czy zmodyfikowano tablicę.

UWAGA

Zaliczenie ćwiczenia odbędzie się na podstawie przesłanego mi mailem pliku Imię_NazwiskoOut.txt

Plik *Imię_NazwiskoOut.txt* proszę przesłać na adres wch@agh.edu.pl z następującym tematem:

„Socjologia II, *Nazwisko Imię*, godz. ..., ”;

Gdzie godz. to rzeczywista godzina zajęć.

Przykłady - polecenia SQL

Polecenie	Składnia ²	Przykład
Logowanie	mysql [-h localhost] [-u student] [-p]	<ul style="list-style-type: none"> mysql -u student -p
Listing baz danych	SHOW DATABASES;	<ul style="list-style-type: none"> SHOW DATABASES;
Tworzenie bazy	CREATE DATABASE moja_baza;	<ul style="list-style-type: none"> CREATE DATABASE cwiczenia;
Wykonanie skryptu	SOURCE nazwa_skryptu.sql;	<ul style="list-style-type: none"> SOURCE d:/student/cwiczenia.sql;
Wypełnienie tabeli danymi z pliku	LOAD DATA LOCAL INFILE 'plik.txt' INTO TABLE moja_tabela;	<ul style="list-style-type: none"> LOAD DATA LOCAL INFILE 'd:/student/abc.txt' INTO TABLE pracownicy;
Wybór bazy danych	USE moja_baza;	<ul style="list-style-type: none"> USE cwiczenia;
Listing tabel w bazie	SHOW TABLES;	<ul style="list-style-type: none"> SHOW TABLES;
Opis pól tabeli	DESCRIBE moja_tabela;	<ul style="list-style-type: none"> DESCRIBE pracownicy;
Tworzenie tabeli	CREATE TABLE moja_tabela ((kolumna typ_danych [NOT NULL NULL] [DEFAULT wartość] [AUTO INCREMENT] [UNIQUE [KEY] PRIMARY [KEY]]), ...);	<ul style="list-style-type: none"> CREATE TABLE pracownicy (imie CHAR(20), nazwisko CHAR(20), wiek INT(3), stanowisko CHAR(20), pensja DEC(9,2));
Dodawanie nowego rekordu	INSERT INTO moja_tabela [(kolumna, ...)] VALUES (wartość, ...);	<ul style="list-style-type: none"> INSERT INTO pracownicy(imie, nazwisko, wiek, pensja) VALUES ('Jan', 'Nowak', '22', '2200');
Aktualizacja rekordów	UPDATE moja_tabela SET kolumna = wartość, ... [WHERE wyrażenie];	<ul style="list-style-type: none"> UPDATE pracownicy SET wiek = 23 WHERE imie = 'Jan' AND nazwisko = 'Nowak';
Usuwanie rekordów	DELETE FROM moja_tabela [WHERE wyrażenie];	<ul style="list-style-type: none"> DELETE FROM pracownicy WHERE imie = 'Jan' AND nazwisko = 'Nowak';
Usuwanie tabeli	DROP TABLE moja_tabela;	<ul style="list-style-type: none"> DROP TABLE pracownicy;
Usuwanie bazy	DROP DATABASE moja_baza;	<ul style="list-style-type: none"> DROP DATABASE cwiczenia;
Selekcja	SELECT [ALL DISTINCT] {kolumna_s formuła_s}, ... [FROM moja_tabela [WHERE wyrażenie_w] [GROUP BY {kolumna_b formuła_b}, ... [ASC DESC]] [HAVING wyrażenie_h] [ORDER BY {kolumna_o formuła_o}, ... [ASC DESC]]];	<ul style="list-style-type: none"> SELECT * FROM pracownicy; SELECT imie, nazwisko FROM pracownicy WHERE Nazwisko LIKE 'N%' AND Imie = 'Jan'; SELECT nazwisko, wiek FROM pracownicy WHERE wiek > 19 ORDER BY nazwisko; SELECT stanowisko, COUNT(*) FROM pracownicy GROUP BY stanowisko HAVING COUNT(*) > 1;

Typy danych

Typy danych	Opis
CHAR (m)	znakowy o stałej długości (m)
VARCHAR (m)	znakowy o zmiennej długości (m)
TEXT	znakowy, długość: 65535
DECIMAL (m, d)	liczba dziesiętna, m – maksymalna liczba cyfr, d – maksymalna liczba cyfr po przecinku
INT	liczba całkowita
DATE	Data

² W opisie składni zastosowano: „[]” – element opcjonalny, „{}” – element obowiązkowy, „|” – alternatywa.