

# BAZY DANYCH



*Lab 1  
Wprowadzenie do  
MySQL-a*

# Laboratorium nr 1

## Temat: Wprowadzenie do MySQL-a

**MySQL** – popularny system do tworzenia i zarządzania bazami danych. Oprogramowanie posiada wysoką prędkość przetwarzania baz danych i zmniejsza koszty inicjacji systemu lub zarządzania bazami danych. Korzystanie z serwera wewnętrznego użytkownik ma możliwość dodawania MySQL w samodzielne oprogramowanie. Oprogramowanie obsługuje także wiele rodzajów tabel, które zapewniają wyszukiwania pełnotekstowego i transakcji na poziomie indywidualnym.

Główne cechy:

- zarządzanie bazą danych,
- duża prędkość przetwarzania baz danych,
- wsparcie dla różnych typów tabelach.

Za każdym razem, kiedy potrzebna jest informacja z bazy danych, należy "zapytać" system bazy danych w zrozumiałym dla niego języku. Tym językiem jest SQL.

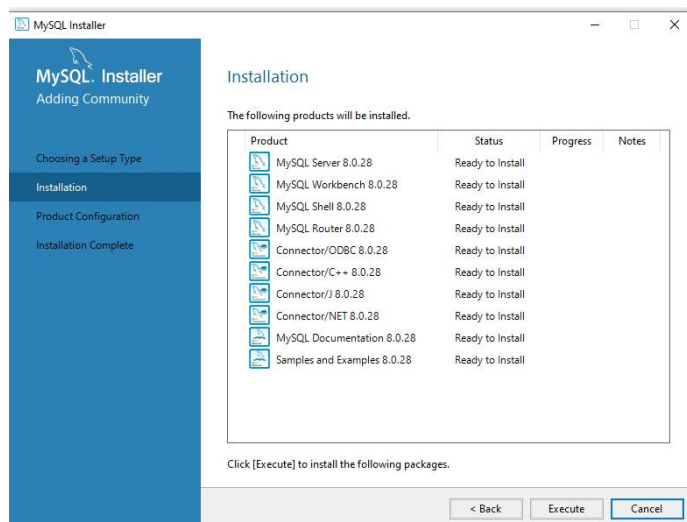
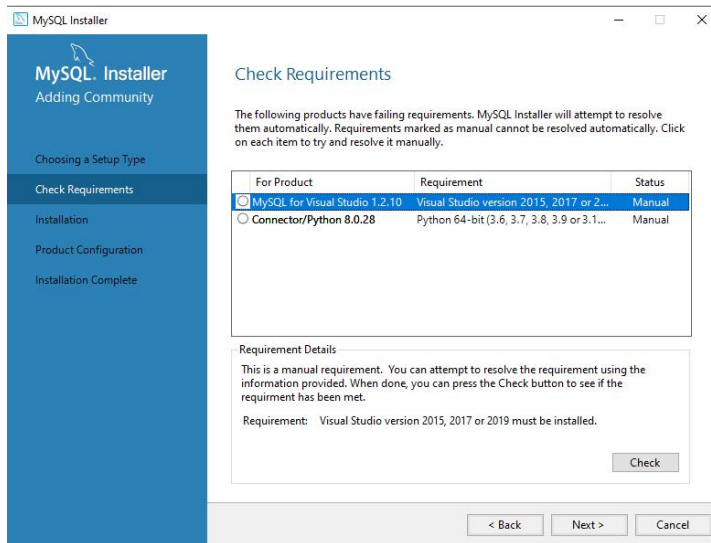
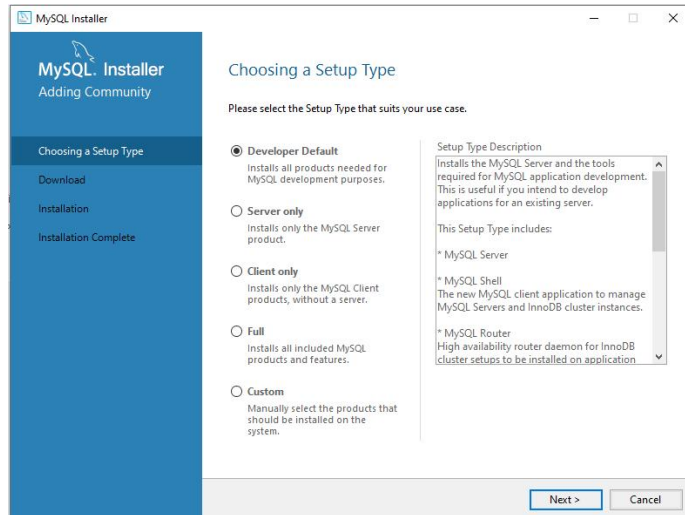
**Język SQL** (ang. *Structured Query Language*) służy do manipulowania danymi umieszczonymi w relacyjnych bazach danych. Jest językiem uniwersalnym, dzięki czemu praca na różnych systemach baz danych sprowadza się do wydawania tych samych lub podobnych komend tzw. zapytań SQL. Język SQL został zaimplementowany w większości relacyjnych systemów baz danych takich jak: DB2, Oracle, InterBase, **MySQL**, dBase, Paradox.

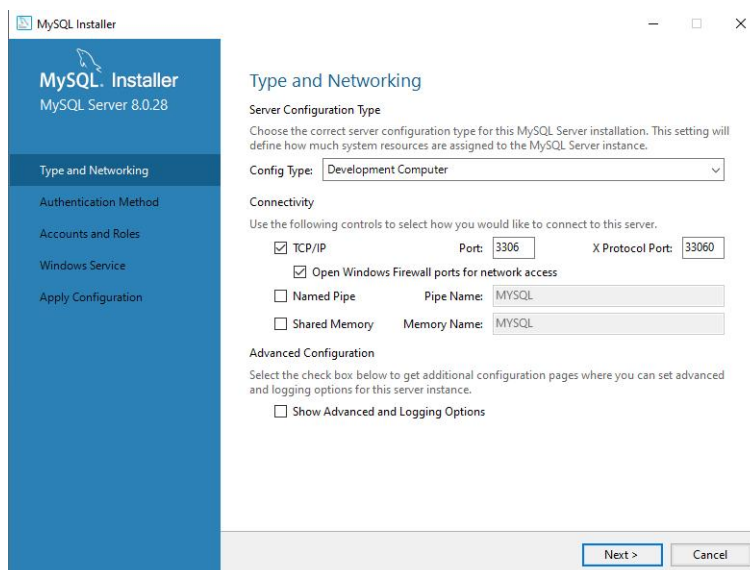
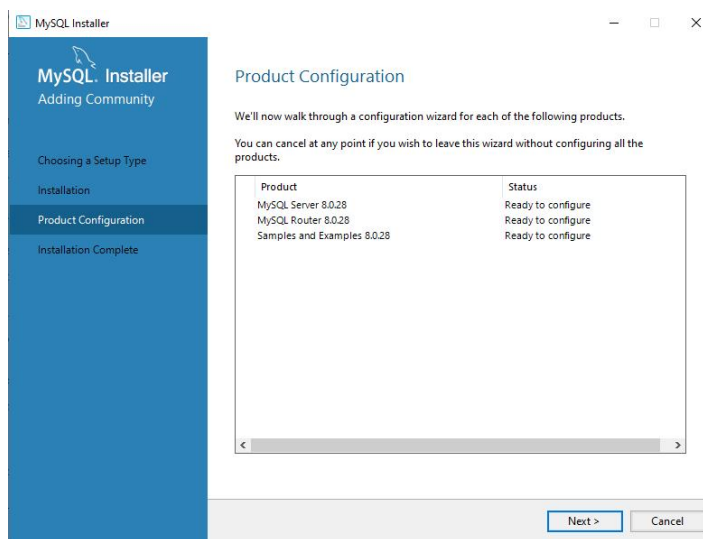
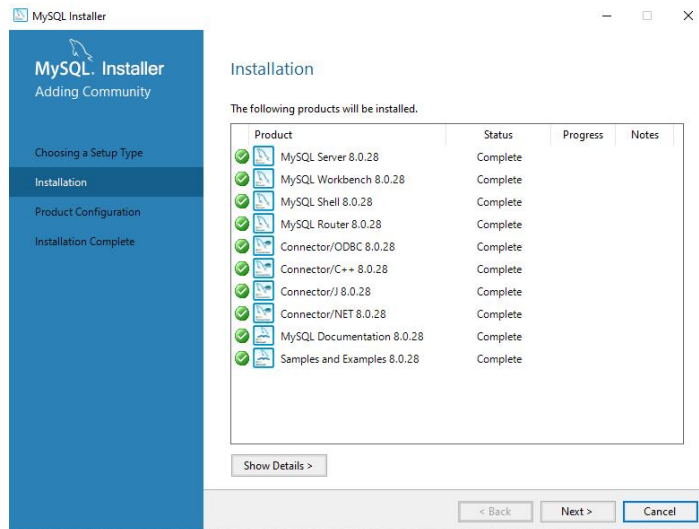
## Instalacja i konfiguracja MySql-a w systemie Windows.

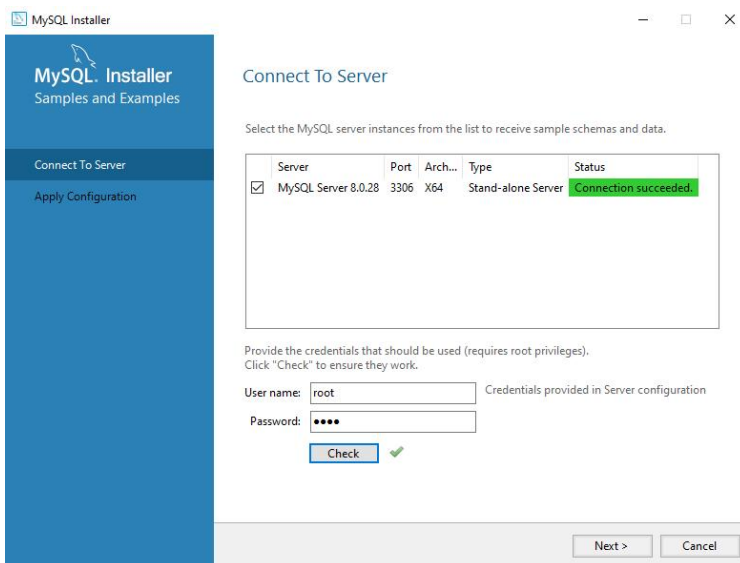
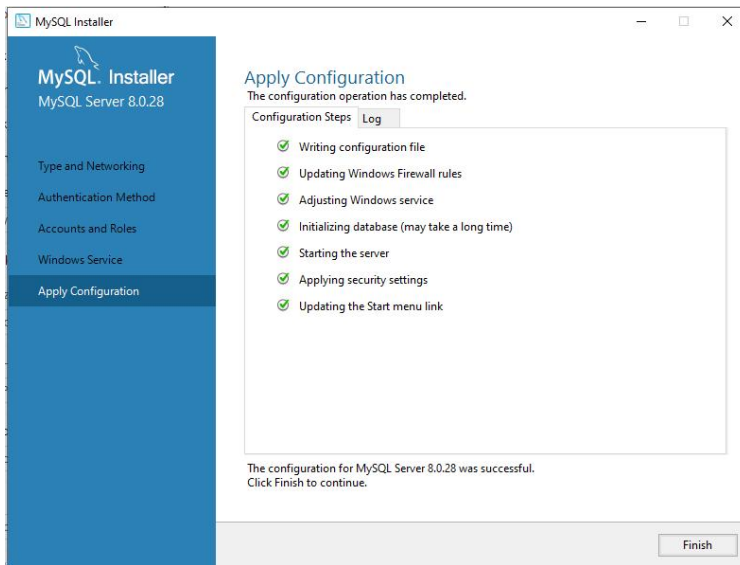
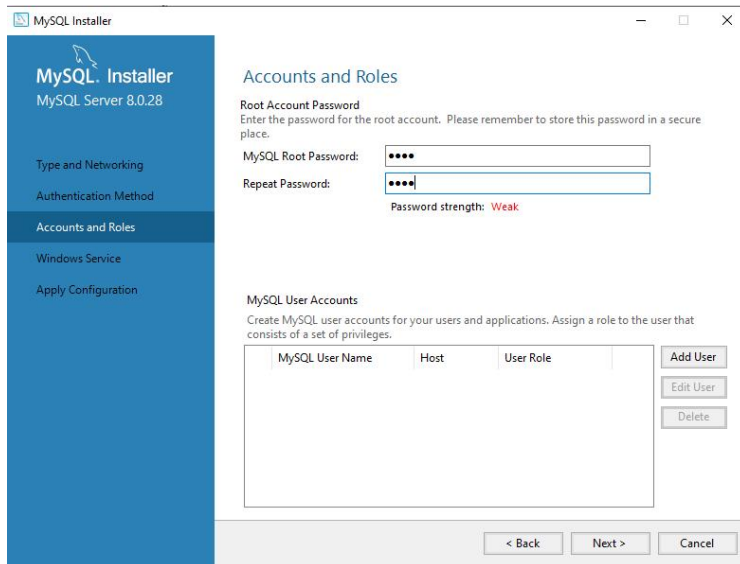
Do zadań przedstawionych w laboratoriach wykorzystana jest wersja MySQL Community Server 8.0.28

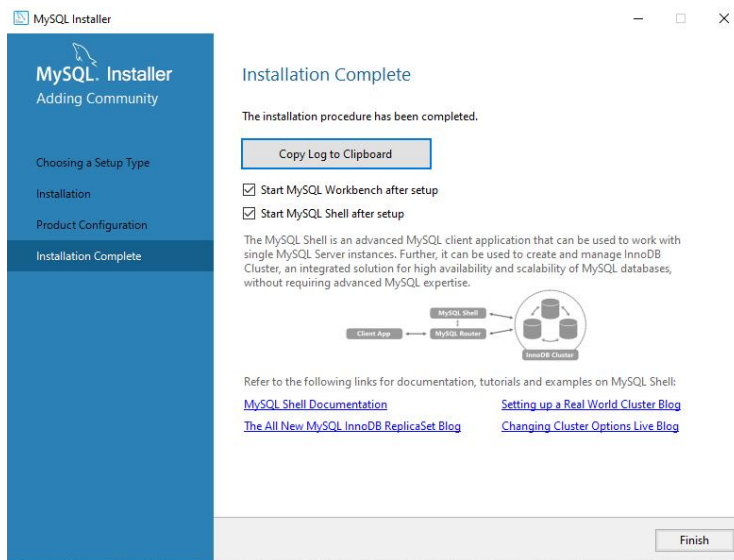
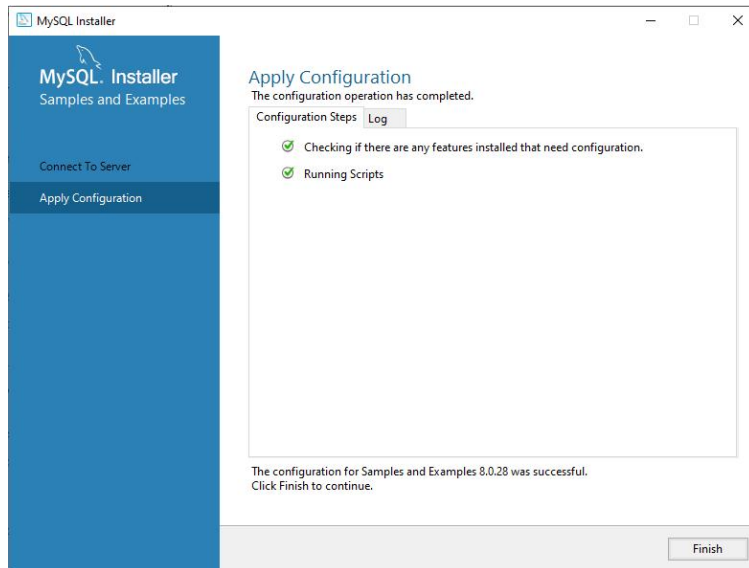
Instalacja przeprowadzana jest za pomocą kreatora. Nie wymaga żadnych dodatkowych ustawień. Po instalacji konieczne jest przeprowadzenie konfiguracji MySQLa w systemie Windows, która także przeprowadzana jest za pomocą kreatora.





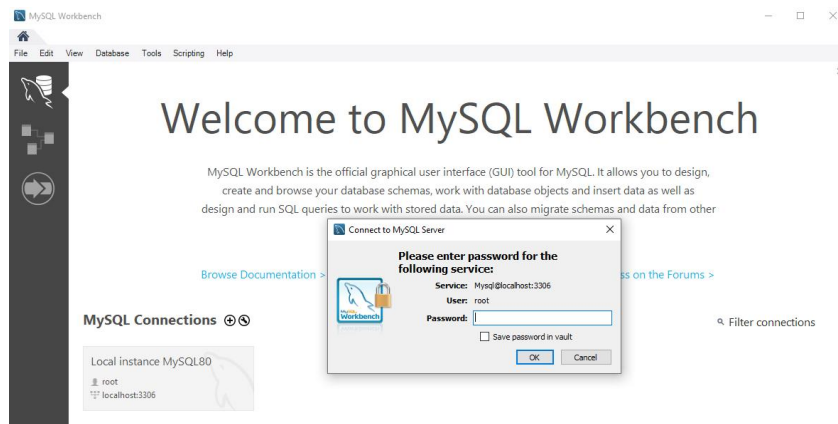






# Praca z bazą danych

Do pracy z programem służy MySQL Workbench



**MySQL Workbench** to środowisko umożliwiające zaprojektowanie bazy danych. Program posiada czytelną szatę graficzną z intuicyjnie rozmieszczonymi narzędziami. Jedną z cech jest możliwość konfigurowania serwera oraz komponentów. Są tu również opcje tworzenia graficznych reprezentacji tabel i innych elementów.

Workbench wspiera procesy stosowane w inżynierii wstecznej, projektowanie koncepcji oraz układu logicznego bazy danych czy importowanie i eksportowanie modeli DBDesigner4. Obsługiwany jest również silnik OpenGL. Pracę ułatwiają dostępne edytory tabel, widoków, procedur, grup. Dane można wyeksportować do formatów PNG, SVG, PDF, PostScript oraz SQL.

Dodatkowo MySQL Workbench posiada zestaw funkcjonalności do zarządzania uprawnieniami, przeglądania logów i raportów. Dostępne jest także narzędzie do tworzenia kopii zapasowych, monitorowania zmiennych i analizy obciążenia konfiguracji sprzętowej.

Produkt wykazuje dużą automatyzacją wielu czynności np. generowanie kodu SQL czy automatyczny zapis plików. Orientację w kodzie ułatwia kolorowanie składni oraz wyszukiwarka.

Z poziomu DOSa można także zainstalować i uruchomić serwer:

Aby serwer był gotowy do pracy należy go zainstalować poleceniem:

```
mysqld --install
```

a następnie uruchomić:

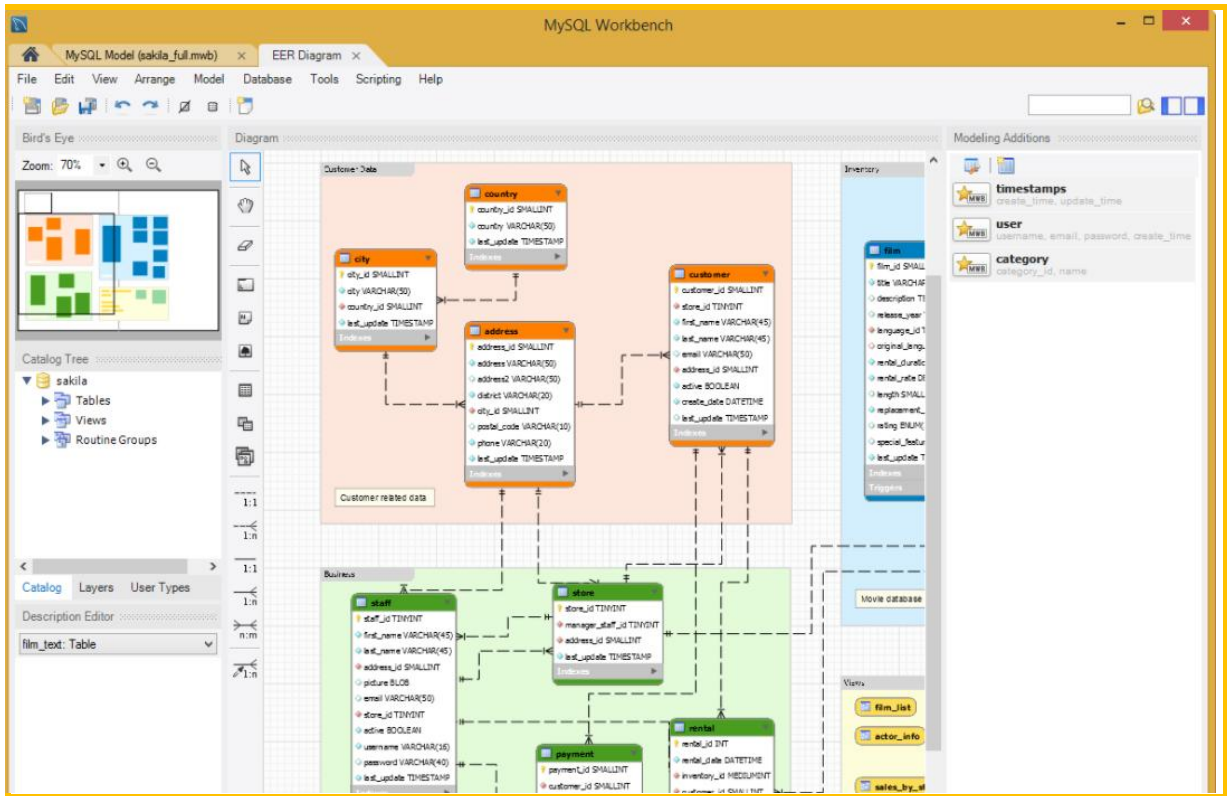
```
net start mysql
```

Proces MySQL będzie działał w tle.

Należy otworzyć wiersz poleceń

Start → Uruchom → cmd.exe

Start → Programy → MySQL → MySQL Server 8.0 → MySql Command Line Client





## Ćwiczenie 1

### Wprowadzenie do programu MySQL

- a) Wyświetl dostępne na serwerze bazy danych.

W celu wyświetlenia listy baz danych należy uruchomić klienta i nawiązać połączenie z bazą a następnie użyć polecenia

```
show databases;
```

- b) Wyświetl dostępne na serwerze bazy danych, których nazwy zaczynają się od słowa mysql.

Aby uzyskać nazwy baz zaczynających się na określony ciąg znaków należy użyć klauzuli LIKE i wzorca mysql%

```
show databases like 'mysql%';
```

- c) Wybierz aktualną bazę do pracy (aby pracować na konkretnej bazie, należy wybrać ją z listy poleceniem USE)

```
use nazwa_bazy_danych;
```

- d) Po wybraniu bazy można sprawdzić jakie tabele się w niej znajdują

```
show tables;
```

- e) Informacje o dowolnej tabeli uzyskujemy poleceniem

```
describe nazwa_tabeli;
```

- f) Wylogowanie z programu następuje po wydaniu komendy

```
\q
```

- Lista poleceń konsoli mysql:

- help (\h) Display this help.
- ? (\?) Synonym for `help`.
- clear (\c) Clear command.
- connect (\r) Reconnect to the server. Optional arguments are db and host.
- edit (\e) Edit command with \$EDITOR.
- ego (\G) Send command to mysql server, display result vertically.
- exit (\q) Exit mysql. Same as quit.
- go (\g) Send command to mysql server.
- no pager (\n) Disable pager, print to std out.
- no tee (\t) Don't write into outfile.
- Pager (\P) Set PAGER [to\_pager]. Print the query results via PAGER.
- print (\p) Print current command.
- prompt (\R) Change your mysql prompt.
- quit (\q) Quit mysql.
- rehash (\#) Rebuild completion hash.
- source (\.) Execute a SQL script file. Takes a file name as an argument.
- status (\s) Get status information from the server.
- system (\!) Execute a system shell command.
- tee (\T) Set out file [to\_outfile]. Append everything into given out file.
- use (\u) Use another database. Takes database name as argument.

- Typy danych w mysql:

- TINYINT[(length)] [UNSIGNED] [ZEROFILL]

- SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
- MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
- INT[(length)] [UNSIGNED] [ZEROFILL]
- INTEGER[(length)] [UNSIGNED] [ZEROFILL]
- BIGINT[(length)] [UNSIGNED] [ZEROFILL]
- REAL[(length, decimals)] [UNSIGNED] [ZEROFILL]
- DOUBLE[(length, decimals)] [UNSIGNED] [ZEROFILL]
- FLOAT[(length, decimals)] [UNSIGNED] [ZEROFILL]
- DECIMAL(length, decimals) [UNSIGNED] [ZEROFILL]
- NUMERIC(length, decimals) [UNSIGNED] [ZEROFILL]
- DATE
- TIME
- TIMESTAMP
- DATETIME
- CHAR(length) [BINARY | ASCII | UNICODE]
- VARCHAR(length) [BINARY]
- TINYBLOB
- BLOB
- MEDIUMBLOB
- LONGBLOB
- TINYTEXT [BINARY]
- TEXT [BINARY]
- MEDIUMTEXT [BINARY]
- LONGTEXT [BINARY]
- ENUM(value1,value2,value3,...)
- SET(value1,value2,value3,...)

## Ćwiczenie 2

### Projektowanie i tworzenie baz danych w MySQL

Tworzenie bazy

```
create database nazwa_bazy;
```

Wykorzystanie bazy danych

```
use nazwa_bazy ;
```

Tworzenie tabel

```
create table nazwa_tabeli (definicja tabeli);
```

np. tabela Pracownicy. Należy zauważyć, że wartość ID\_pracownika nie może być pusta, dlatego zostało to zaznaczone już na etapie tworzenia tabeli NOT NULL)

Przykłady:

```
CREATE TABLE Stanowiska(
id_stanowisko INTEGER PRIMARY KEY NOT NULL,
nazwa          VARCHAR(40));
```

```

CREATE TABLE Pracownicy (
  Id_pracownika      INTEGER PRIMARY KEY NOT NULL,
  Nazwisko           VARCHAR(35),
  Imie               VARCHAR(15),
  Stanowisko         INTEGER,
  Data_zatrudnienia DATE,
  FOREIGN KEY FK_PR_RELA_STANOW (Stanowisko)
    references STANOWISKA (Id_stanowisko));

```

Klucz główny **primary key** jest to kolumna lub zestaw kolumn , którego będziemy używać do identyfikowania pojedynczego rekordu tabeli .Tworzymy podając na końcu tabeli odpowiednie wskazanie. Wyznaczamy parametr który ma być naszym kluczem głównym.

Klucze obce **foreign key** reprezentują związki między tabelami. Obie kolumny tabel, na których zakładany jest klucz obcy muszą być tego samego typu. Znak i rozmiar dla typu integer muszą być identyczne, ale nie muszą być identyczne łańcuchy tekstowe, na których zakładany jest klucz obcy.

Określenie takiego związku między tabelą Pracownicy a Stanowiska będzie wyglądało następująco (obie tabele muszą być wcześniej określone w bazie):

```

alter table Pracownicy
  add foreign key FK_PR_RELA_STANOW (Stanowisko)
    references STANOWISKA (Id_stanowisko) on delete restrict;

```

Drugim sposobem na wprowadzenie klucza obcego jest zdefiniowanie go na etapie definicji struktury tabeli

```

...
  Data_zatrudnienia DATE,
  FOREIGN KEY FK_PR_RELA_STANOW (Stanowisko)
    references STANOWISKA (Id_stanowisko));

```

Proszę utworzyć bazę danych (MySQL) na podstawie schematu i zdefiniować potrzebne ograniczenia i klucze.

