



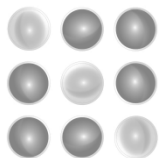
KAPITAŁ LUDZKI  
NARODOWA STRATEGIA SPÓJNOŚCI



UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



kwalifikacje  
dla rynku pracy

Politechnika Lubelska przyjazna dla pracodawcy



**Opracowanie dydaktycznych materiałów do przedmiotów na studiach  
stacjonarnych I stopnia kierunku Informatyka w ramach realizacji projektu  
"Kwalifikacje dla rynku pracy - Politechnika Lubelska przyjazna dla  
pracodawcy"**

Opracowanie dydaktycznych materiałów do laboratoriów do  
przedmiotu „**Inżynieria systemów baz danych**” na kierunku  
Informatyka  
(I stopień kształcenia)

*dr inż. Maria Skublewska-Paszkowska*

Politechnika Lubelska 2014

**Człowiek – najlepsza inwestycja**

### Zagadnienia laboratorium

*Laboratorium 1* – projektowanie struktury bazy danych, dostosowanych do przedstawionego problemu/opisu działania przedsiębiorstwa

*Laboratorium 2* – projektowanie struktury bazy danych w środowisku bazodanowym MySQL, tworzenie diagramu EER, tworzenie skryptów bazy danych

*Laboratorium 3* – uzupełnianie danymi bazy danych w systemie MySQL

*Laboratorium 4* – zarządzanie bazą danych MySQL

*Laboratorium 5* – tworzenie bazy danych w systemie PostgreSQL

*Laboratorium 6* – wstawianie danych do tabel bazy danych w systemie PostgreSQL

*Laboratorium 7* – zarządzanie bazą danych PostgreSQL

*Laboratorium 8* – tworzenie bazy danych w systemie MS SQL

*Laboratorium 9* – wstawianie danych do tabel bazy danych MS SQL

*Laboratorium 10* – zarządzanie bazą danych MS SQL

*Laboratorium 11* – programowanie w języku T-SQL

## **Inżynieria baz danych**

### ***Laboratorium 1***

Modelowanie struktury relacyjnej bazy danych dostosowanego do podanych wymagań

#### **Cel zajęć**

---

Umiejętność projektowania struktury baz danych na podstawie zadanego opisu

Umiejętność wyłonienia tabel i ich pól, a także definiowanie relacji zachodzących między nimi

Umiejętność analizowania utworzonej struktury bazy danych

Umiejętność przekształcania struktury bazy danych ze względu na niepoprawne relacje

#### **Wymagania wstępne**

---

Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami encji.

#### **Zadanie:**

Na podstawie poniższego opisu należy przeanalizować potrzeby przedsiębiorstwa dotyczące przechowywanych informacji i ich późniejszego wykorzystania. Należy wyłonić przyszłych użytkowników systemu informatycznego, a także ich role w firmie. Należy wypisać encje i przydzielić do nich atrybuty. Należy zastanowić się nad kluczami. Należy opisać każdy atrybut względem jej właściwości.

#### **Przykładowy opis działania firmy:**

*Rozpatrywana firma prowadzi prywatną szkołę językową dla dzieci i młodzieży. Szkoła posiada główną siedzibę, ale także wiele filii rozlokowanych w różnych częściach kraju (nawet kilka filii w danym mieście). W szkole zajęcia językowe prowadzone są przez danych nauczycieli, których dane są przechowywane w systemie. Każda filia posiada sekretariat do obsługi placówki. Zajęcia językowe podzielone są według rodzajów kursów. Powinna istnieć możliwość dodawania nowych rodzajów ze względu na modyfikacje programów nauczania. Uczestnicy szkoły podzieleni są według kursów, ale także według zajęć w obrębie danego*

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*kursu. Istnieje ustalony grafik zajęć, kursantów, który jest ustalony na dany rok. W danej grupie może uczestniczyć maksymalnie 12 osób.*

*Pracownicy dostają miesięczne wynagrodzenie, o którym informacja także jest przechowywana w systemie. Każdy nauczyciel zatrudniony jest: na umowę o pracę lub umowę o dzieło.*

*Uczestnicy wnoszą opłatę z kurs w ratach rozłożonych według schematu: albo miesięcznych, kwartalnych, półrocznych lub w postaci jednorazowej opłaty.*

*Do każdego kursu uczestnicy dostają podręczniki oraz płyty (każdy kurs posiada ustaloną liczbę książek oraz płyt).*

*Każdy uczestnik kursu dostaje unikalny login i hasło, który upoważnia zalogowania do strony, a także dostęp do dodatkowych materiałów. Dodatkowo przechowywany jest email użytkownika, na które wysyłane są informacje o zajęciach.*

1. Wyłonienie przyszłych użytkowników systemu dla powyższego opisu:

- Administrator – zarządzanie systemem, najwyższe uprawnienia
- Nauczyciel – prowadzenie zajęć
- Sekretariat – zarządzanie sekretariatem i obsługą filii

2. Wyłonienie listy tabel.

*Przykładowe tabele wyłonione według opisu:*

- Kursanci
- Pracownicy (np. nauczyciel, sekretarka, administrator)
- Kursy
- Rodzaje kursów
- Filie
- Zajęcia
- Płatności(dokonywane przez kursantów)
- Raty płatności
- Wynagrodzenia
- Umowy
- Rodzaje umowy
- Hasła dostępu

### 3. Opis kolejnych tabel

Opis tabel powinien zostać przedstawiony w tabeli. Każda tabela musi zostać szczegółowo opisana.

*Przykładowy opis przedstawiono w tabeli poniżej. Każda tabela powinna zostać zidentyfikowana poprzez nazwę. Należy wypełnić opis tabeli dotyczący jakie dane będzie ona przechowywała. Opis powinien być w miarę dokładny, aby w przyszłości wyłonić podstawowe jej kolumny (atrybuty).*

<b>Nazwa tabeli</b>	<b>Typ</b>	<b>Opis</b>
<i>Kursanci</i>	Dane	Informacje dotyczące danych kursantów w postaci imienia, nazwiska, adresu zamieszkania oraz danych kontaktowych
<i>Filie</i>	Dane	Fizyczne lokalizacje filii rozmieszczonych w różnych miastach w różnych dzielnicach. Zawierają informacje o adresie danej filii oraz wewnętrznym numerze.
<i>Rodzaje kursów</i>	Podzbiór	Zawiera informacje o kursach w postaci nazwy kursu, jego opisu, numeru, liczby książek oraz płyt przypisanych do niego, jak także liczbę godzin kursu.

*Każda tabela ma zidentyfikowany typ. Dostępne jej rodzaje to:*

- *tabela danych – jest reprezentacją obiektu lub podmiotu, istotnego z punktu widzenia instytucji; zawiera kluczowe dane dla bazy danych*
- *tabela łącząca – tabela, która definiuje powiązanie pomiędzy dwoma tabelami w relacji wiele do wielu*
- *tabela typu podzbiór – zawiera pola, które są związane z określoną tabelą danych i bardzo dokładnie opisuje podmiot tabeli*
- *tabela walidacyjna – zawiera dane stosunkowo statystyczne, jest kluczowym składnikiem integralności danych*

#### 4. Wyłonienie pól poszczególnych tabel

Należy określić, które pola najlepiej definiują cechy tabeli, a następnie przypisać je do tabeli. Należy zdecydować przez jaki typ danych będą reprezentowane. W przypadku ciągu znaków należy ustalić maksymalną liczbę znaków. Dodatkowo należy ustalić, które pola są wymagane, a które są opcjonalne.

Przykładowa reprezentacja tabeli Kursanci

<i><b>Nazwa pola</b></i>	<i><b>Typ pola</b></i>	<i><b>Czy pole jest wymagane?</b></i>	<i><b>Czy wartość jest unikatowa?</b></i>	<i><b>Opis</b></i>
<i>IdKursanta (id)</i>	całkowity	TAK	TAK	Wewnętrzny identyfikator klienta
<i>Imię</i>	Znakowe (max. 30)	TAK	NIE	Ciąg określający imię użytkownika
<i>Drugie imię</i>	Znakowe (max. 30)	NIE	NIE	Ciąg określający drugie imię użytkownika
<i>Nazwisko</i>	Znakowe (max. 35)	TAK	NIE	Ciąg określający nazwisko użytkownika
<i>Województwo</i>	Znakowe (max 35)	TAK	NIE	Ciąg znaków określający województwo w Polsce
<i>Miasto</i>	Znakowe (max 50)	TAK	NIE	Ciąg znaków określający miasto lub miejscowość w której jest zameldowany użytkownik
<i>Ulica</i>	Znakowe (max. 50)	TAK	NIE	Ciąg znaków określający nazwę ulicy

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

<i>Nr</i>	Znakowe (max. 5)	TAK	NIE	Ciąg znaków określający numer bloku lub domu
<i>Nr Mieszkania</i>	Znakowe (max. 5)	NIE	NIE	Ciąg znaków określający numer mieszkania lub lokalu
<i>Kod pocztowy</i>	Znakowy ( 5)	TAK	NIE	Ciąg znaków określający kod pocztowy w postaci pięciu liczb
<i>Email</i>	Znakowy (max. 30)	TAK	NIE	Ciąg znaków określający email w odpowiedniej postaci
<i>Telefon</i>	Znakowy (max. 12)	NIE	NIE	Ciąg znaków określający numer telefonu
<i>PESEL</i>	(znakowy – 11)	TAK	TAK	Ciąg znaków określający pesel- 11 liczb

*Należy upewnić się, że wszystkie wyróżnione pola są polami prostymi, a nie złożonymi (składającymi się z wielu połączonych informacji). Takie pola powodują problemy przy pobieraniu danych z bazy.*

*Należy wyeliminować wszystkie duplikaty pól – niepotrzebnie dana wartość przechowywana w wielu tabelach.*

## 5. Definiowanie kluczy w tabelach

Należy zidentyfikować i przypisać klucze kandydujące do każdej tabeli.

*Klucze kandydujące* – pole lub zestaw pól, które w jednoznaczny sposób pojedynczą instancję tabeli. Każda tabela musi posiadać co najmniej jeden klucz kandydujący.

*Przykładowo dla tabeli Kursanci można wymienić klucze takie jak: IdKursanta, Pesel,*

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*nr telefonu. Przykładem klucza kandydującego złożonego mogą być powiązane pola: Nazwisko oraz Imię.*

*Kolejnym zadaniem należy wybrać klucze główne dla każdej tabeli. Klucz główny:*

- *identyfikuje tabelę w strukturze bazy danych, odgrywa ważną rolę podczas tworzenia relacji pomiędzy tabelami*
- *identyfikuje każdy rekord i zapewnia jego unikalność, a ponadto zabezpiecza przed wstawianiem duplikowanych rekordów.*

<b>Nazwa tabeli</b>	<b>Nazwa klucza głównego</b>
Kursanci	IdKursanta
Pracownik	IdPracownika
Filia	IdFilii
Kursy	IdKursu
Zajecia	IdZajecia
Płatnosci	IdPłatnosi

Należy zdefiniować klucze obce dla wszystkich tabel.

Przykład kluczy obcych dla tabeli *Kursy* przedstawia poniższa tabela:

<b>Nazwa tabeli pierwotnej</b>	<b>Nazwa klucza obcego</b>
Rodzaje Kursów	IdRodzajuKursu
Filie	IdFilii

Przykład kluczy obcych dla tabeli *Umowa* przedstawia poniższa tabela:

<b>Nazwa tabeli pierwotnej</b>	<b>Nazwa klucza obcego</b>
Pracownicy	idPracownik
Rodzaje Umów	IdRodzajUmowy

Przykład kluczy obcych dla tabeli *Zajecia* przedstawia poniższa tabela:

<b>Nazwa tabeli pierwotnej</b>	<b>Nazwa klucza obcego</b>
Pracownicy	idPracownik
Kursanci	idKursant
Kursy	idKurs



## 6. Identyfikowanie relacji tabel

Należy zidentyfikować powiązania pomiędzy tabelami i ich licznosc z użyciem macierzy wszystkich tabel.

*Przykład macierzy dla wybranych kolumn został przedstawiony poniżej:*

	<b>Pracownicy</b>	<b>Kursy</b>	<b>Zajecia</b>	<b>Kursanci</b>	<b>RodzajeKursow</b>
<b>Pracownicy</b>			1:N		
<b>Kursy</b>			1:1		1:N
<b>Zajecia</b>	1:1			1:N	
<b>Kursanci</b>		1:N	1:N		
<b>RodzajeKursow</b>		1:1			

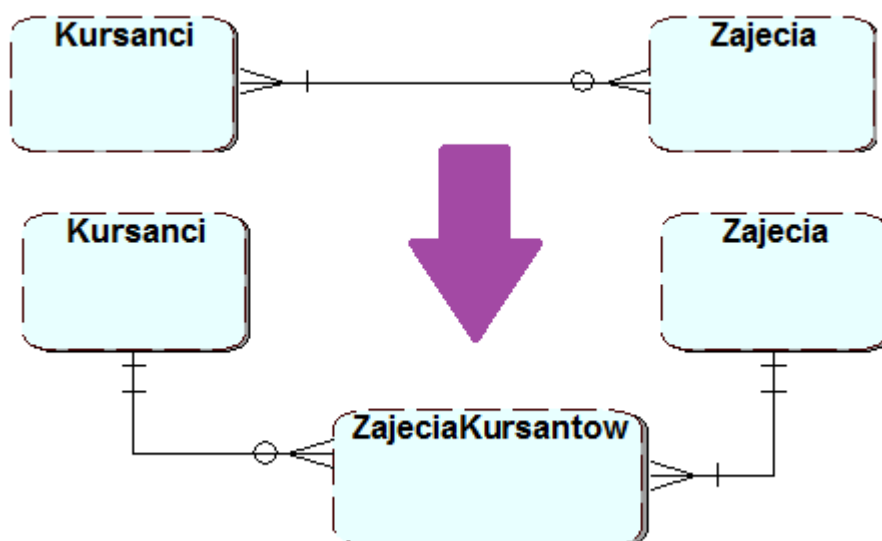
## 7. Identyfikowanie problematycznych relacji wiele do wielu

Należy przeanalizować strukturę bazy, wyłonić związki wiele do wielu i przebudować strukturę w taki sposób, aby związki te zlikwidować.

*W omawianym przykładzie relacja wiele do wielu zachodzi pomiędzy tabelami Zajecia oraz Kursanci. Każdy Kursant może uczestniczyć w wielu kursach, a na konkretne zajęcia zapisanych jest wielu Kursantów.*

*Ze względu na niedogodności związane z tym typem relacji, należy rozbić ten związek i wstawić pośredniczącą tabelę, przykładowo nazwaną Zajecia Kursantów.*

*Przykład przekształcenia związku wiele do wielu jest przedstawiony na poniższym rysunku.*



*W przedstawionym przykładzie utworzono nową tabelę o nazwie ZajeciaKursantow, która jest typu tabeli łączącej.*

## 8. Zaprojektowanie struktury bazy danych

Na podstawie zgromadzonych informacji należy utworzyć schemat bazy danych (diagram ERD). Na diagramie należy umieścić wszystkie encje (tabele) i ich atrybuty (pola), a także relacje pomiędzy encjami. Należy ustalić odpowiednie typy danych związane z atrybutami.

Każda tabela musi posiadać zidentyfikowany klucz główny. Te tabele, które tego wymagają powinny mieć zidentyfikowany klucz obcy.

## 9. Analiza utworzonej struktury bazy danych

- a. Należy sprawdzić, czy utworzona baza danych posiada strukturę zgodną z opisem przedsiębiorstwa. W przypadku stwierdzonych nieprawidłowości należy dostosować jej strukturę.
- b. Należy sprawdzić, czy każda tabela spełnia warunek tabeli idealnej.

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

- c. Należy zdefiniować reguły usuwania dla wszystkich relacji bazy danych.

Należy zidentyfikować

<i><b>Rodzaj związku</b></i>	<i><b>Reguła usuwania danych z bazy</b></i>

- d. Należy przeprowadzić analizę integralności bazy danych.

## Inżynieria baz danych

### Laboratorium 2

#### Tworzenie bazy danych w systemie MySQL

#### Cel zajęć

Umiejętność obsługi środowiska MySQL Workbench

Umiejętność utworzenia bazy danych w środowisku MySQL Workbench

Umiejętność utworzenia schematu EER

Umiejętność utworzenia skryptu bazodanowego

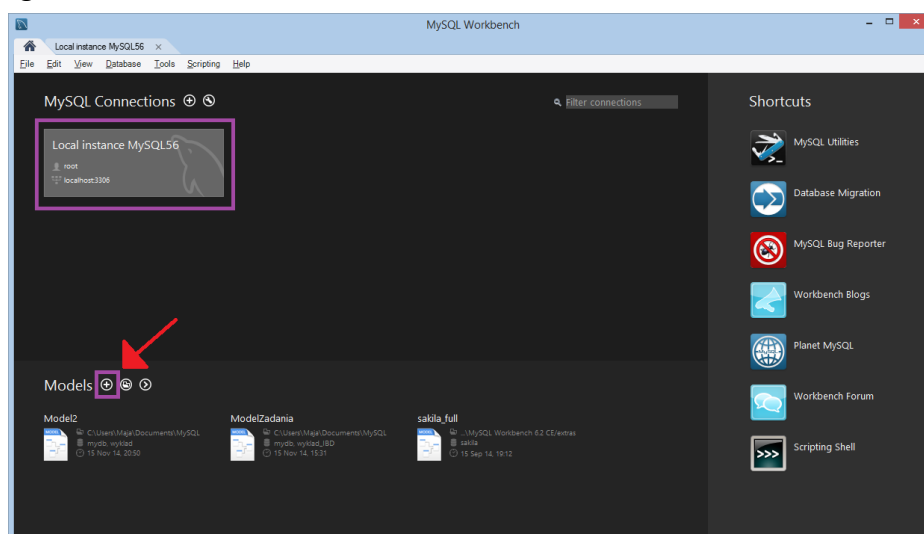
#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami encji.

Student posiada teoretyczną wiedzę dotyczącą środowiska MySQL Workbench i zna jego możliwości.

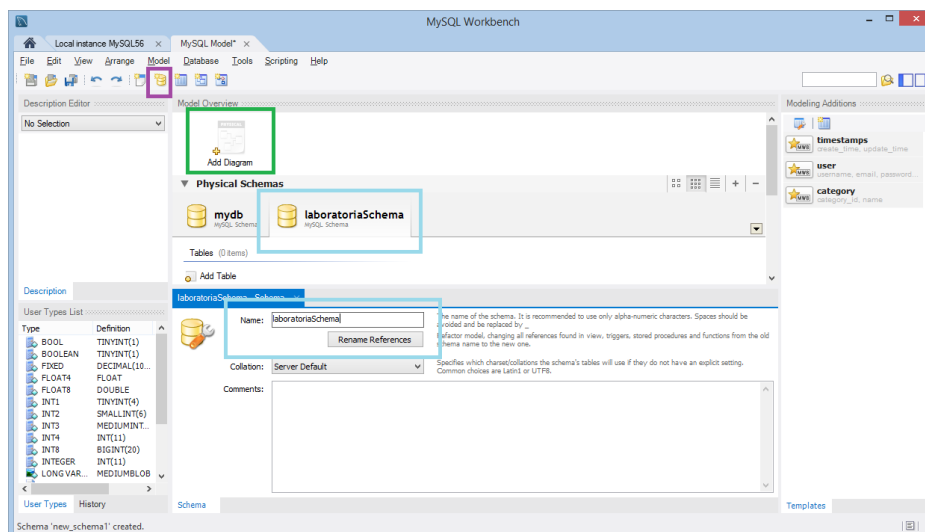
1. Należy utworzyć diagram EER na podstawie zdefiniowanej struktury z poprzednich zajęciach.

Należy uruchomić program MySQL Workbench, połączyć się do bazy i wybrać tworzenie nowego diagramu.



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Należy utworzyć nowy schemat, poprzez wybranie odpowiedniej opcji. Diagram EER ma być utworzony w nowym schemacie.



Na rysunku kolorem fioletowym zaznaczono tworzenie nowego schematu, któremu przypisano nową nazwę. Automatycznie tworzona jest zakładka nowego schematu i ustawiana jest jako domyślna dla nowo tworzonego schematu. Kolorem niebieskim zaznaczono wprowadzane informacje o schemacie.

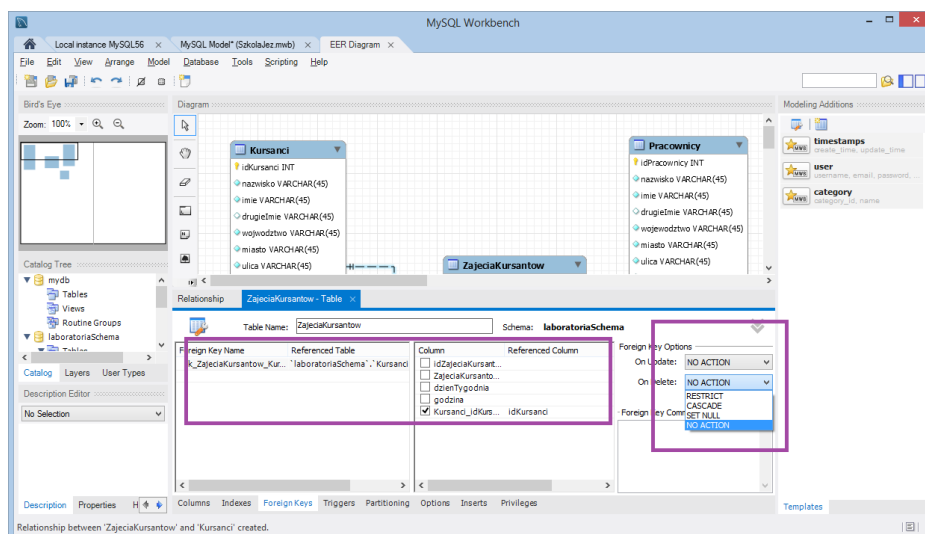
Po naciśnięciu ikony Add diagram (zaznaczonego kolorem zielonym), zostanie utworzony nowy, pusty diagram EER.

Należy utworzyć diagram EER:

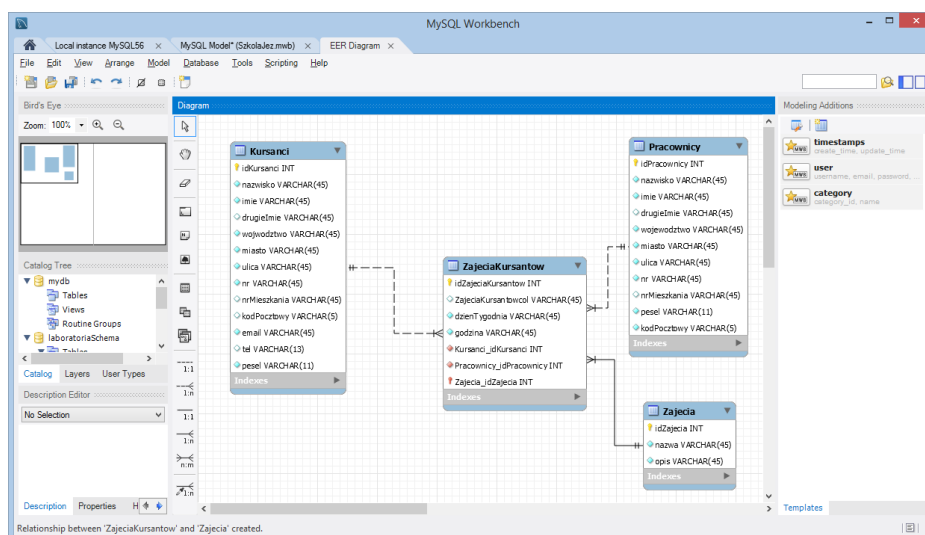
- zdefiniować tabele
- zdefiniować pola tabel i przypisać im odpowiedni typ
- należy zdefiniować klucze główne
- należy zdefiniować relacje między tabelami oraz klucze obce

*Przykład modyfikowania relacji pomiędzy tabelami – Foreign Key.*

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



Przykład części diagramu EER został przedstawiony na poniższym rysunku.



- Należy wygenerować schemat SQL tworzenia bazy danych na podstawie Forward Engineer. Następnie należy przeanalizować skrypt.

Należy utworzyć skrypt SQL poprzez wybranie opcji zmenu Database->Forward Engineer. Następnie należy postępować zgodnie z kolejnymi krokami. W opcjach można zaznaczyć jakie obiekty mają być tworzone. Cały skrypt może zostać zapisany do pliku.

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

**Set Options for Database to be Created**

Tables

- ☐ Skip creation of FOREIGN KEYS
- ☐ Skip creation of FK Indexes as well
- ☐ Generate separate CREATE INDEX statements
- ☒ Generate INSERT statements for tables
- ☐ Disable FK checks for INSERTs

Other Objects

- ☐ Don't create view placeholder tables
- ☐ Do not create users. Only create privileges (GRANTS)

Code Generation

- ☐ DROP objects before each CREATE object
- ☐ Generate DROP SCHEMA
- ☐ Omit schema qualifier in object names
- ☐ Generate USE statements
- ☐ Add SHOW WARNINGS after every DDL statement
- ☒ Include model attached scripts

Back Next Cancel

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

**Select Objects to Forward Engineer**

To exclude objects of a specific type from the SQL Export, disable the corresponding checkbox. Press Show Filter and add objects or patterns to the ignore list to exclude them from the export.

- ☒ Export MySQL Table Objects  
4 Total Objects, 4 Selected [Show Filter](#)
- ☐ Export MySQL View Objects  
0 Total Objects, 0 Selected [Show Filter](#)
- ☐ Export MySQL Routine Objects  
0 Total Objects, 0 Selected [Show Filter](#)
- ☐ Export MySQL Trigger Objects  
0 Total Objects, 0 Selected [Show Filter](#)
- ☐ Export User Objects  
0 Total Objects, 0 Selected [Show Filter](#)

Back Next Cancel

Forward Engineer to Database

Connection Options

Options

Select Objects

Review SQL Script

Commit Progress

**Review the SQL Script to be Executed**

This script will now be executed on the DB server to create your databases.  
You may make changes before executing.

```

13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
-- Schema laboratoriaSchema
--
CREATE SCHEMA IF NOT EXISTS 'laboratoriaSchema' ;
USE 'laboratoriaSchema' ;
-- Table 'laboratoriaSchema'.'Kursanci'
--
CREATE TABLE IF NOT EXISTS 'laboratoriaSchema'.'Kursanci' (
'idKursanci' INT NOT NULL,
'nazwisko' VARCHAR(45) NOT NULL,
'imie' VARCHAR(45) NOT NULL,
'drugieImie' VARCHAR(45) NULL,
'wojwodztwo' VARCHAR(45) NOT NULL,
'miasto' VARCHAR(45) NOT NULL,
'ulica' VARCHAR(45) NOT NULL,
'nr' VARCHAR(45) NOT NULL,
'nrMieszkania' VARCHAR(45) NULL,
'kodPocztowy' VARCHAR(5) NULL,
'email' VARCHAR(45) NOT NULL,
'tel' VARCHAR(45) NULL,
'pesel' VARCHAR(11) NOT NULL,
PRIMARY KEY ('idKursanci'),
UNIQUE INDEX 'email_UNIQUE' ('email' ASC),
UNIQUE INDEX 'tel_UNIQUE' ('tel' ASC), ..

```

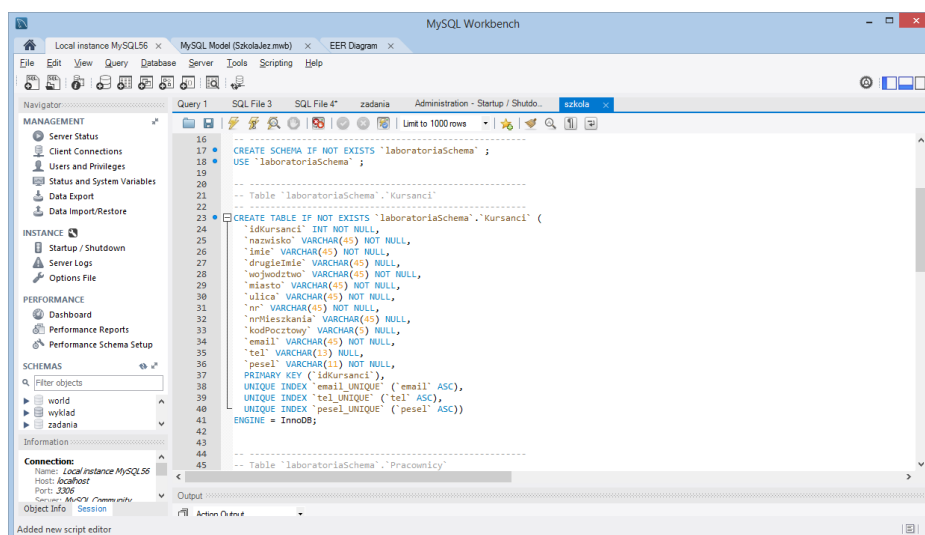
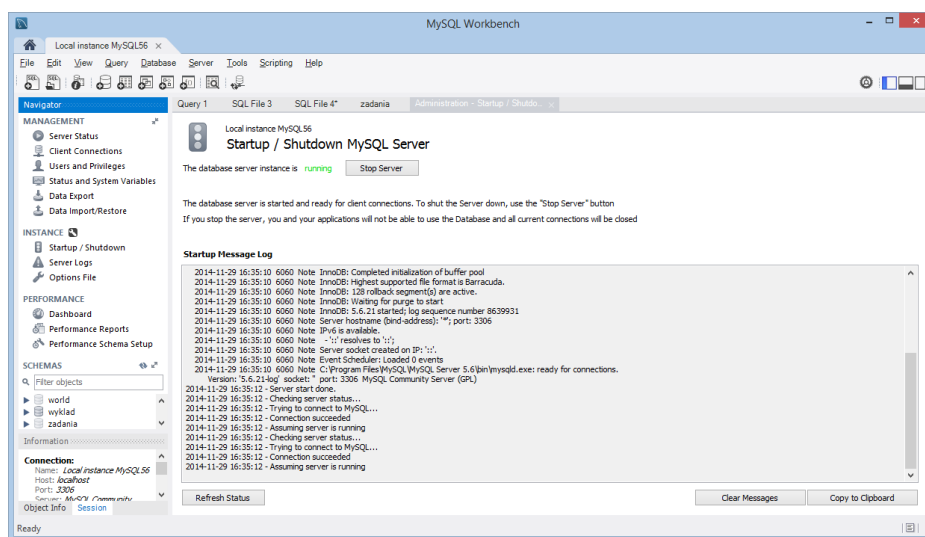
Save to File... Copy to Clipboard

Back Next Cancel

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

### 3. Na podstawie wygenerowane skryptu z zadania 3 należy utworzyć tabele w bazie danych.

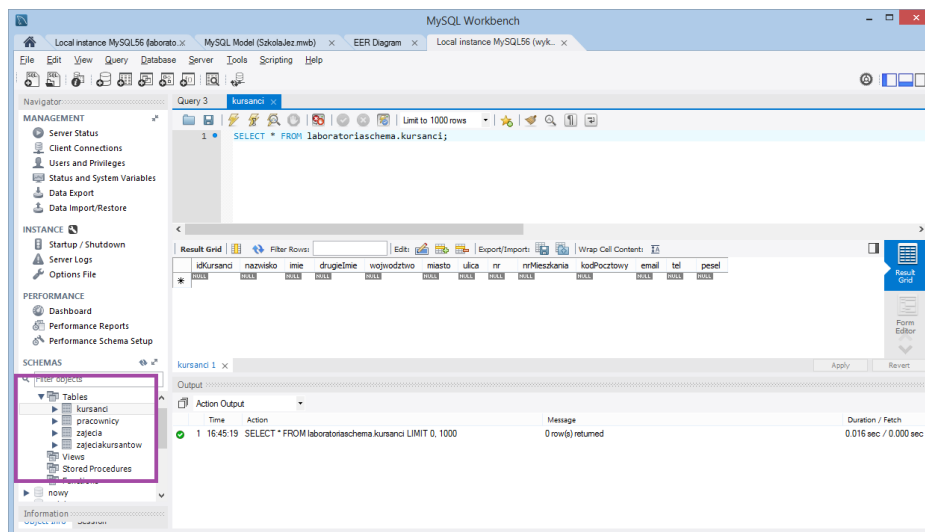
*W tym celu należy wybrać „Open a SQL script file in a new query tab”, a następnie wybrać nazwę skryptu, który należy uruchomić. Zostanie otwarta nowa zakładka z kodem SQL, który należy wykonać. W logach można sprawdzić, czy operacje przebiegły pomyślnie.*



*Po wykonaniu operacji nowy schemat i wszystkie tabele zostały utworzone, co prezentuje poniższy rysunek.*



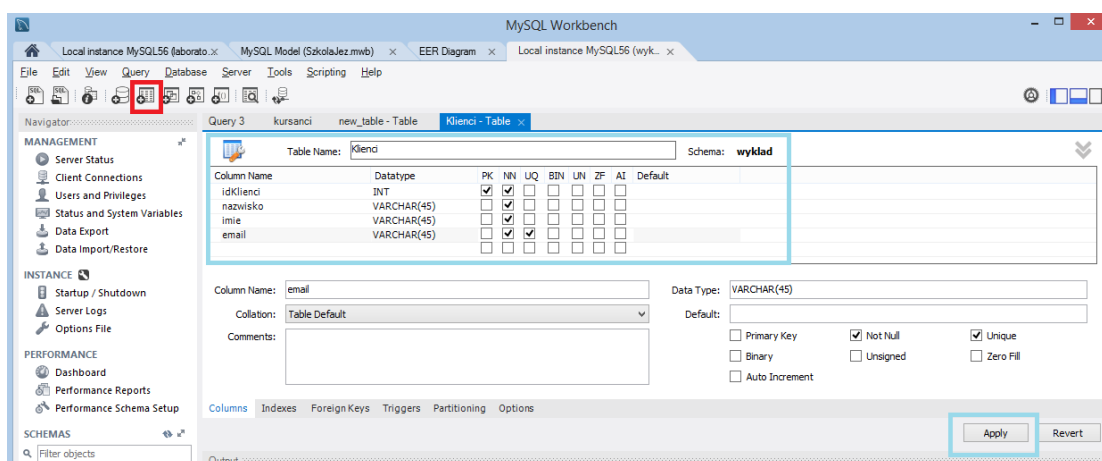
Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



- Należy utworzyć ręcznie trzy tabele w bazie danych, ustawić klucze główne, a także ustawić relacje je łączące.

Trzy tabele: Klienci, Zamówienia, ZamówieniaKlienta.

Należy wybrać opcję dodawania nowej tabeli, a następnie wypełnić kolejne jej pola.



- Należy napisać prosty schemat skryptu SQL tworzący bazę danych opartych na kilku tabelach. Na podstawie skryptu należy utworzyć diagram bazy danych.

Skrypt ma dotyczyć utworzenia bazy danych składający się z trzech tabel: Klienci, Zamówienia, ZamówieniaKlienta.

W tym celu należy z menu wybrać: Database-> Reverse Engineer

## Inżynieria baz danych

### Laboratorium 3

Uzupełnianie danymi bazy danych w systemie MySQL

#### Cel zajęć

Umiejętność obsługi środowiska MySQL Workbench

Umiejętność dodawania danych do tabel bazy danych

Umiejętność utworzenia skryptu wprowadzającego dane do tabel

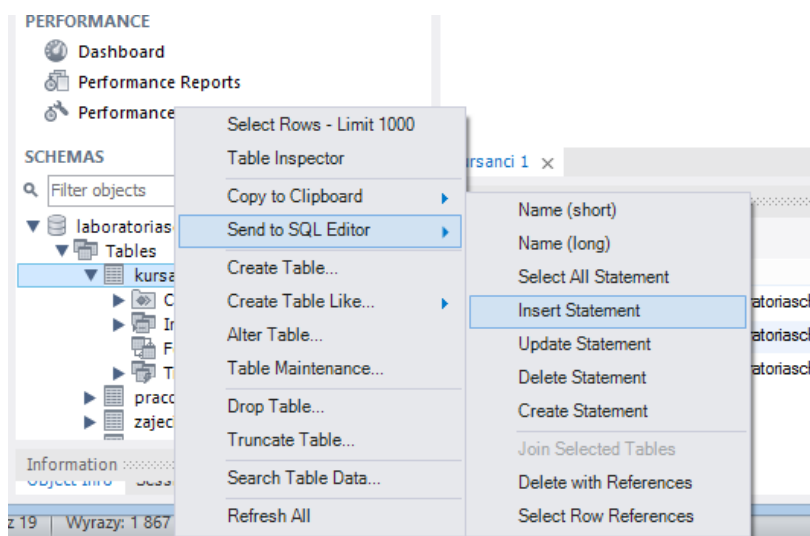
#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami encji.

Student posiada teoretyczną wiedzę dotyczącą środowiska MySQL Workbench i zna jego możliwości.

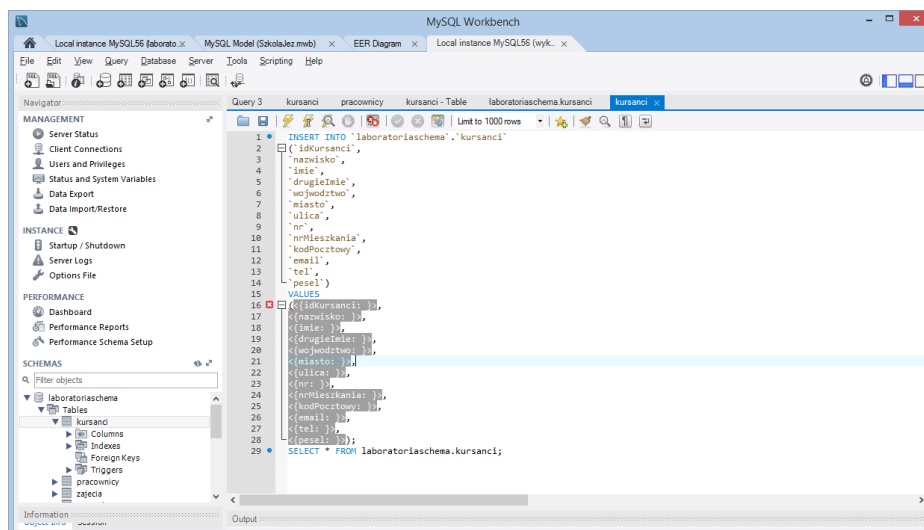
1. Należy wprowadzić nowy rekord do tabeli *Kursanci*.

Z listy tabel po lewej stronie należy wybrać tabelę *Kursanci*, wybrać z rozwijanego menu *Send to SQL Editor-> Insert Statement*. Zostało to przedstawione na poniższym rysunku.



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Zostanie otwarty edytor SQL ze schematem polecenia wprowadzania danych dla wybranej tabeli. Przykład dla tabeli Kursanci został przedstawiony poniżej.*



*W poleceniu INSERT INTO podany jest schemat, a także należąca do niego nazwa tabeli. W pierwszej części zapytania wymienione są wszystkie kolumny tabeli. W drugiej części są przygotowane miejsca do wpisania wartości wprowadzanych do tabeli.*

*Jeśli wprowadzane dane są zgodne z typami danych, polecenie zostanie poprawnie wykonane, a dane zostaną wprowadzone do bazy danych. Jeśli dana wartość nie jest wymagana i zostanie opuszczona, należy albo podać wartość NULL lub pominąć wartość i odnoszącą się do niej nazwę kolumny.*

*Przykład zapytania przedstawiono poniżej.*

```
INSERT INTO `laboratoriaschema`.`kursanci`
(`idKursanci`,
`nazwisko`, `imie`, `wojwodztwo`, `miasto`, `ulica`, `nr`, `nrMieszkania`,
`kodPocztowy`, `email`, `tel`, `pesel`)
VALUES
(1, 'Kowalska', 'Monika', 'Lubelskie', 'Lublin', 'Nadbyst', '24',
'125A', '20815', 'am@wp.pl',
'44577899', '91123456554');
```

2. Należy wprowadzić po jednym rekordzie do pozostałych tabel w utworzonej bazie danych.

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

3. Należy utworzyć skrypt SQL wprowadzający po jednym rekordzie do każdej tabeli bazy danych w narzędziu MySQL Workbench. Utworzony skrypt należy zapisać do pliku.

*W edytorze SQL należy wprowadzić polecenia tworzenia nowych rekordów dla tabel i zapisać je jako skrypt SQL. Pozwoli to na łatwiejsze wprowadzanie danych w późniejszym czasie.*

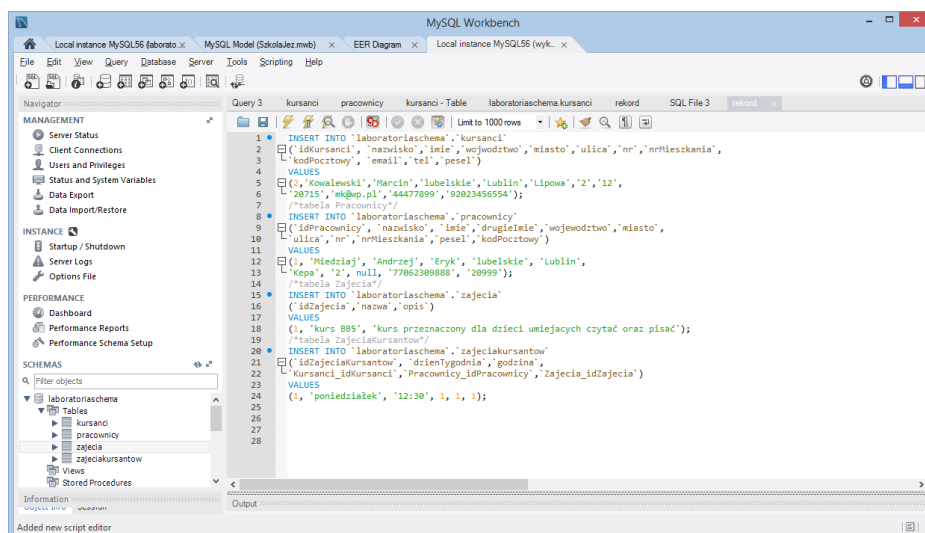
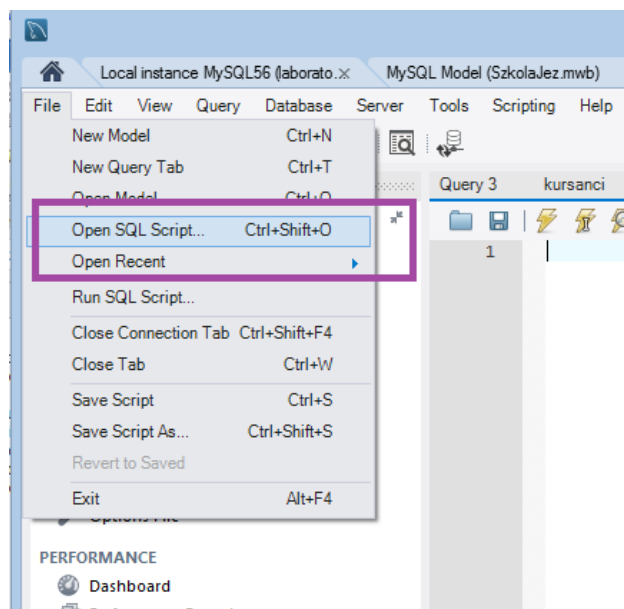
*Aby zapisać skrypt należy wybrać z menu Plik->Save script (lub Plik->Save script as).*

*Utworzony skrypt powinien zawierać polecenia, które przedstawiono poniżej.*

```
/*tabela Kursanci*/
INSERT INTO `laboratoriaschema`.`kursanci`
(`idKursanci`, `nazwisko`, `imie`, `wojwodztwo`, `miasto`, `ulica`, `nr`, `nrMieszkania`,
`kodPocztowy`, `email`, `tel`, `pesel`)
VALUES
(2, 'Kowalewski', 'Marcin', 'lubelskie', 'Lublin', 'Lipowa', '2', '12',
'20715', 'mk@wp.pl', '44477899', '92023456554');
/*tabela Pracownicy*/
INSERT INTO `laboratoriaschema`.`pracownicy`
(`idPracownicy`, `nazwisko`, `imie`, `drugieImie`, `wojewodztwo`, `miasto`,
`ulica`, `nr`, `nrMieszkania`, `pesel`, `kodPocztowy`)
VALUES
(1, 'Miedziaj', 'Andrzej', 'Eryk', 'lubelskie', 'Lublin',
'Kepa', '2', null, '77062309888', '20999');
/*tabela Zajecia*/
INSERT INTO `laboratoriaschema`.`zajecia`
(`idZajecia`, `nazwa`, `opis`)
VALUES
(1, 'kurs BB5', 'kurs przeznaczony dla dzieci umiejących czytać oraz pisać');
/*tabela ZajeciaKursantow*/
INSERT INTO `laboratoriaschema`.`zajeciakursantow`
(`idZajeciaKursantow`, `dzienTygodnia`, `godzina`,
`Kursanci_idKursanci`, `Pracownicy_idPracownicy`, `Zajecia_idZajecia`)
VALUES
(1, 'poniedziałek', '12:30', 1, 1, 1);
```

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

- Należy wczytać zapisany skrypt, zmodyfikować go i ponownie uruchomić. W tabelach powinny zostać wstawione kolejne rekordy.



## Inżynieria baz danych

### Laboratorium 4

#### Zarządzanie bazą danych MySQL

#### Cel zajęć

Umiejętność obsługi środowiska MySQL Workbench

Umiejętność tworzenia nowych użytkowników

Umiejętność tworzenia eksportu oraz importu bazy danych

Umiejętność wykonywania kopii zapasowej

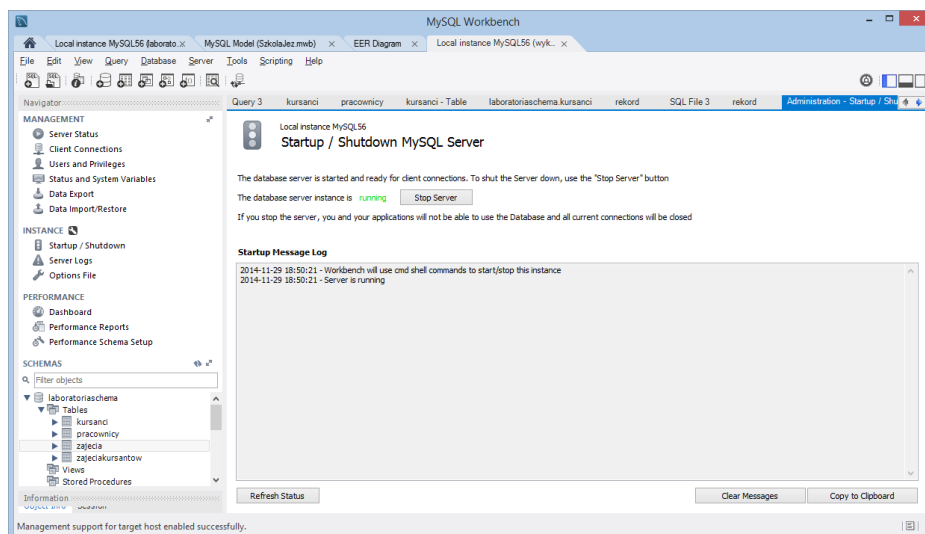
#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą środowiska MySQL Workbench i zna jego możliwości.

1. Należy sprawdzić status połączenia bazy danych

*W tym celu należy wybrać informacje o instancji bazy danych Startup/Shutdown. Jeśli instancja nie jest wyłączona, należy ją włączyć.*

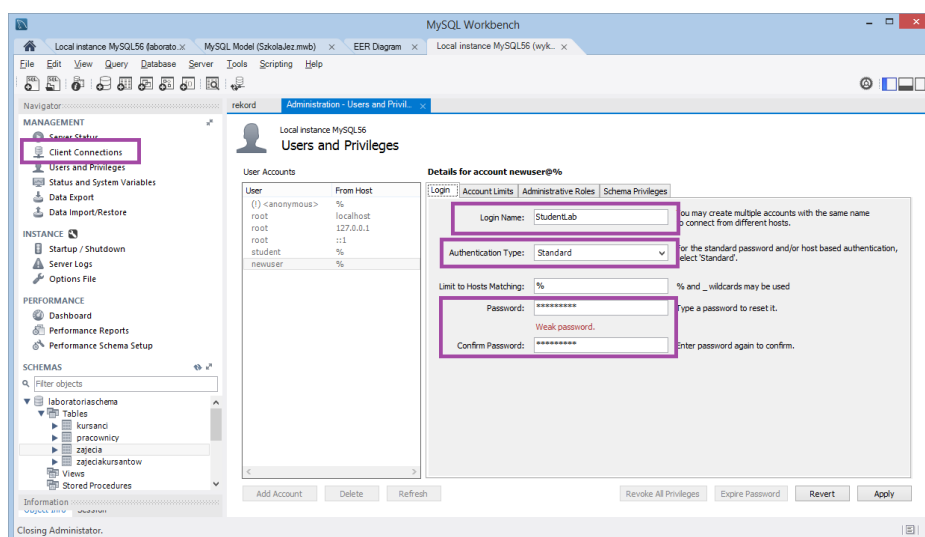
*Na poniższym rysunku widać informację o działającej instancji.*



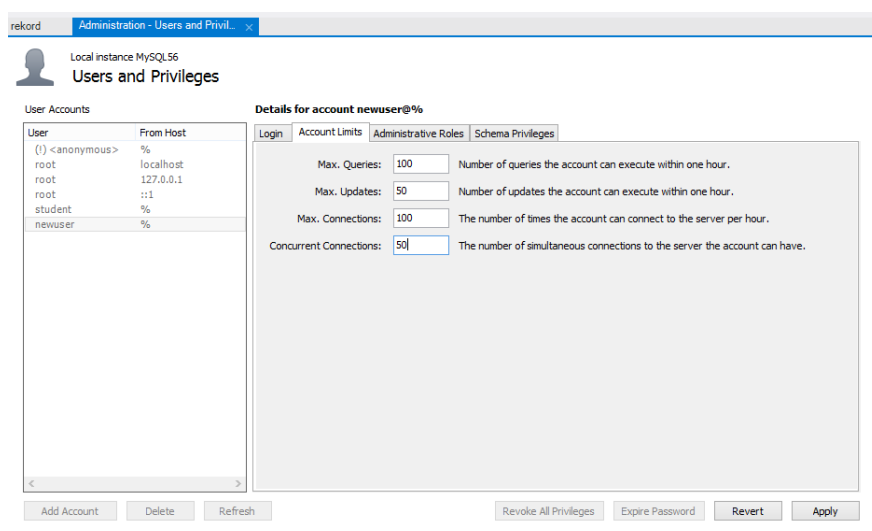
Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

- Należy utworzyć nowego użytkownika o nazwie *StudentLab*. Jako typ identyfikacji należy wybrać opcję *Standard*. Należy podać nowe hasło dla użytkownika (np. *student14*).

Nowego użytkownika można utworzyć z użyciem narzędzia *MySQL Workbench* w sekcji zarządzającej (*Management*), *Users and Privileges*. Zostanie otwarte nowe okno, przedstawione poniżej, w które należy wpisać nazwę użytkownika, rodzaj autentyfikacji oraz hasło.

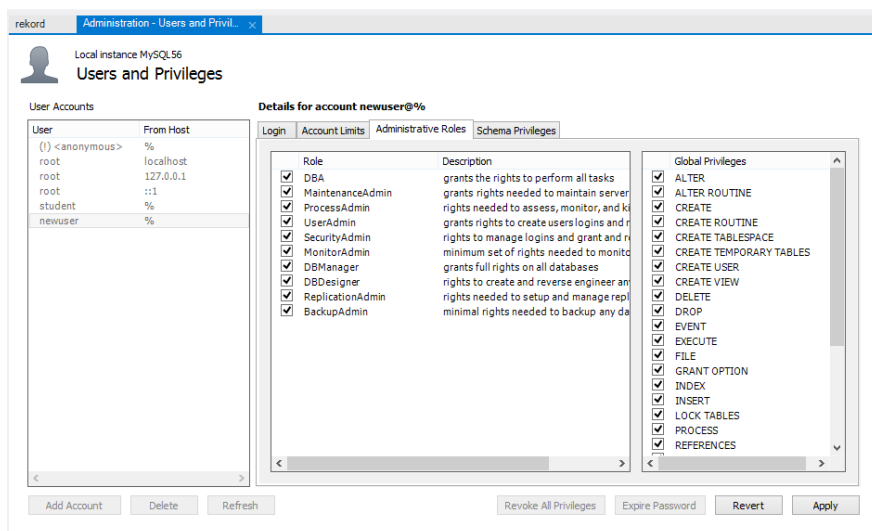


W drugiej zakładce można ustawić ograniczenia przypisane do konta (głównie godzinowe). Poniżej przedstawiono przykładowe wartości. Domyślnie pola mają wpisane zera.

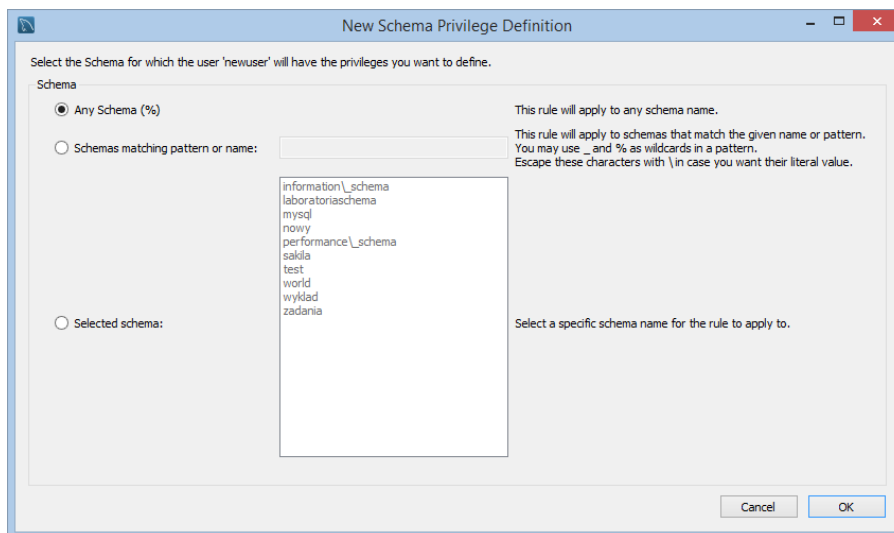


Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Kolejna zakładka dotyczy uprawnień użytkownika. Należy wybrać odpowiednie opcje. W przypadku uprawnień administratora (DBA), użytkownik uzyskuje wszystkie uprawnienia.*



*Istnieje możliwość przypisania uprawnień użytkownika do wszystkich schematów lub tylko do wybranych. Domyślnie jest zastosowana do wszystkich. Na poniższym oknie przedstawiono możliwość wyboru.*



*Po przyciśnięciu przycisku Apply zostanie utworzony nowy użytkownik.*

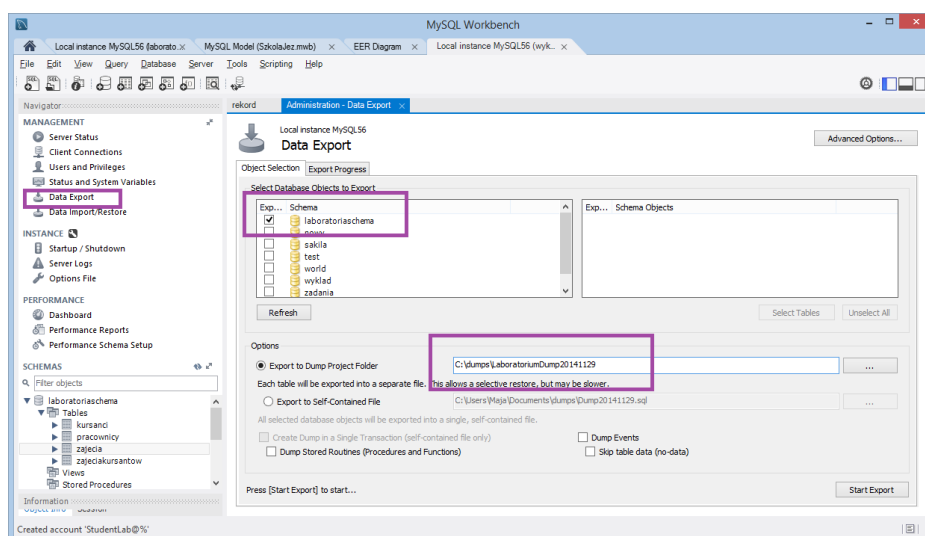


Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

### 3. Należy wykonać eksport bazy danych.

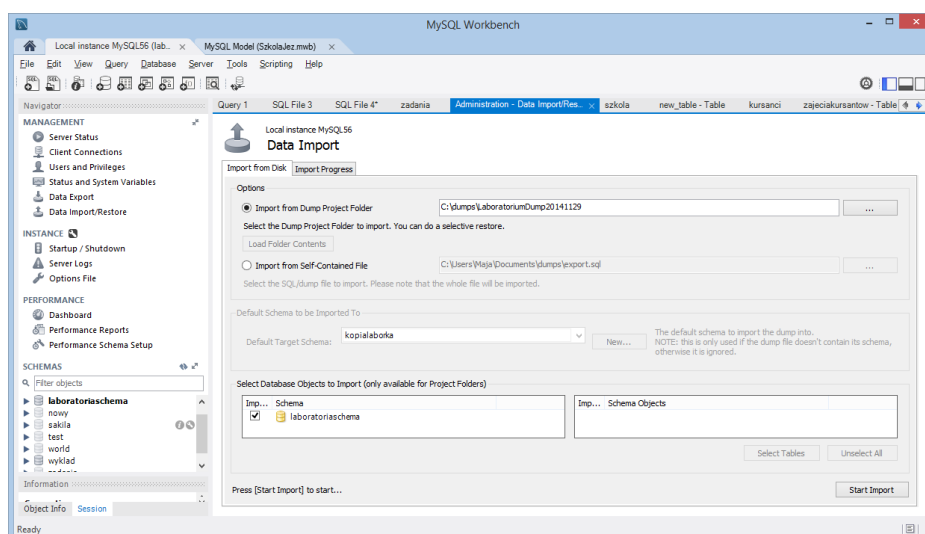
*Wykonywanie eksportu bazy danych jest ważny z punktu widzenia utrzymania danych. Eksport pozwala na zapisanie danych i struktury bazy danych do pliku. Ten plik może zostać użyty do tworzenia nowej bazy danych.*

*W celu wykonania eksportu bazy danych należy wybrać Dataexport, a następnie postępować kolejno według instrukcji. Wybór schematu do eksportu jest przedstawiony na poniższym rysunku.*



### 4. Należy usunąć bazę danych, a następnie wykonać import bazy danych

*Możliwość przywracania bazy danych jako całej lub wybranej tabeli. Można wybrać domyślny schemat bazy danych.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Wykonywanie kopii zapasowej można przeprowadzić poleceniem `mysqldump` w powłoce. Należy podać nazwę użytkownika, który ma uprawnienia wykonywania polecenia `SELECT` do danej bazy danych (lub wszystkich). Należy podać nazwę bazy danych oraz nazwę pliku, w którym będzie przechowywana kopia zapasowa.*

## Inżynieria baz danych

### Laboratorium 5

#### Tworzenie bazy danych w systemie PostgreSQL

##### Cel zajęć

Umiejętność obsługi środowiska pgAdmin

Umiejętność utworzenia bazy danych w narzędziu pgAdmin

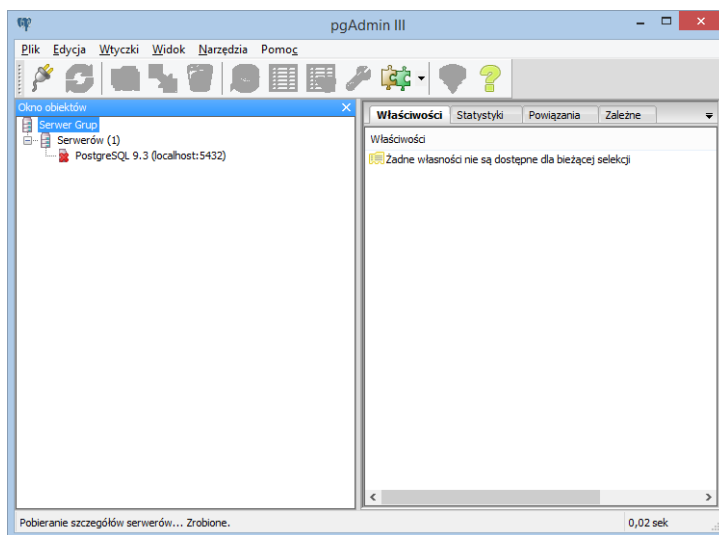
Umiejętność utworzenia tabel, ręcznie i z użyciem skryptu SQL

##### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami a encjami.

Student posiada teoretyczną wiedzę dotyczącą środowiska pgAdmin i zna jego możliwości.

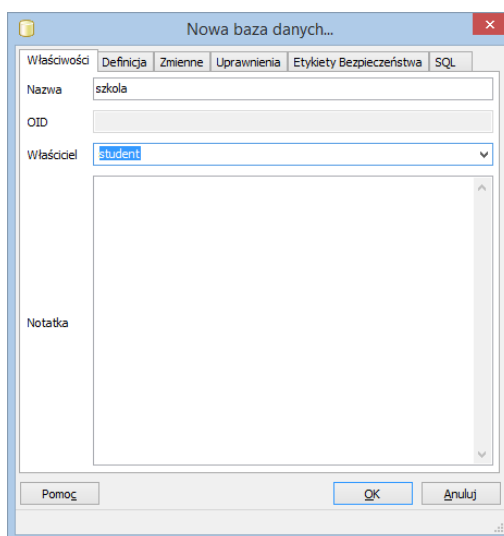
1. Należy otworzyć narzędzie pgAdmin i podłączyć się do bazy danych



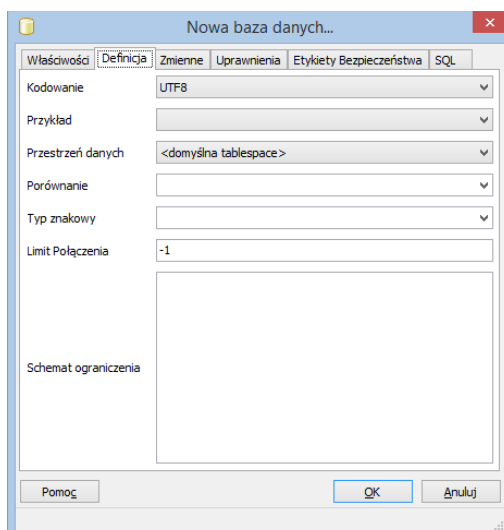
Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

2. Należy utworzyć nową bazę danych, zgodnie z opracowanymi wymaganiami opracowanymi na pierwszym laboratorium. Bazie należy nadać nazwę.

*Należy utworzyć nową bazę danych klikając prawym przyciskiem myszy na drzewie Bazy. Pojawi się okno z zakładkami definiującymi bazę danych. Poniższa zakładka definiuje nazwę i właściciela, wybieranego z listy wszystkich dostępnych.*

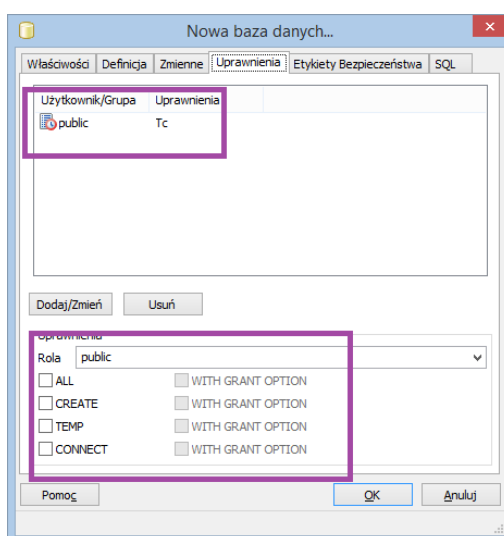


*Druga zakładka dotyczy definicji dla tworzonej bazy danych. Wybierane jest kodowanie (domyślnie UTF8), przestrzeń tabel, porównanie oraz typ znakowy. Dodatkowo ustawiana jest liczba połączeń do bazy danych. Domyślnie ustawiana jest wartość -1 co oznacza brak limitu.*

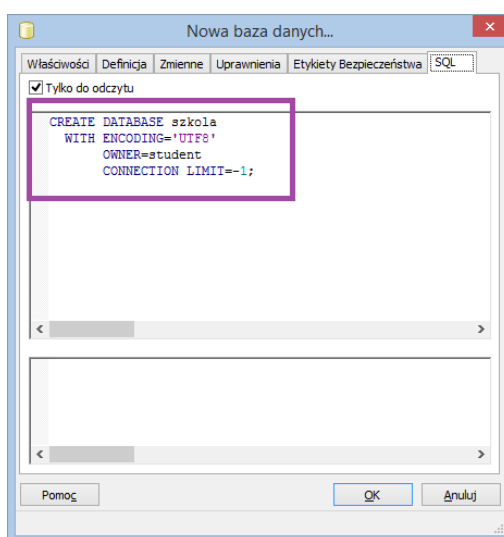


Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Czwarta zakładka dotyczy uprawnień związanych z bazą danych. W pierwszej części jest grupa z uprawnieniami. Poniżej można zaznaczyć dane uprawnienia do bazy danych (ALL – wszystkie uprawnienia zostają przyznane – należy tę opcję zaznaczyć). Domyślnie zaznaczona jest rola public. Jeśli istnieją zdefiniowane inne role w bazie, można je wybrać z listy rozwijanej.



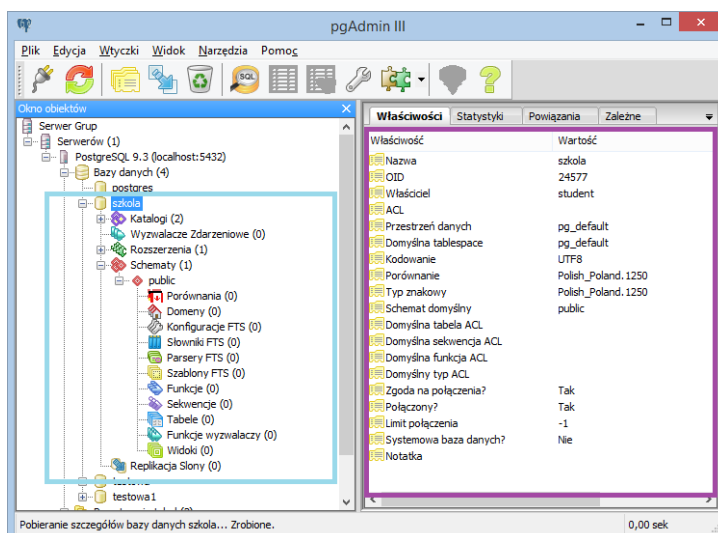
W ostatniej zakładce widoczny jest kod SQL, który został stworzony na podstawie wybranych ustawień.



Po zapisaniu wszystkich ustawień, zostanie utworzona nowa baza danych. Ustawienia

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

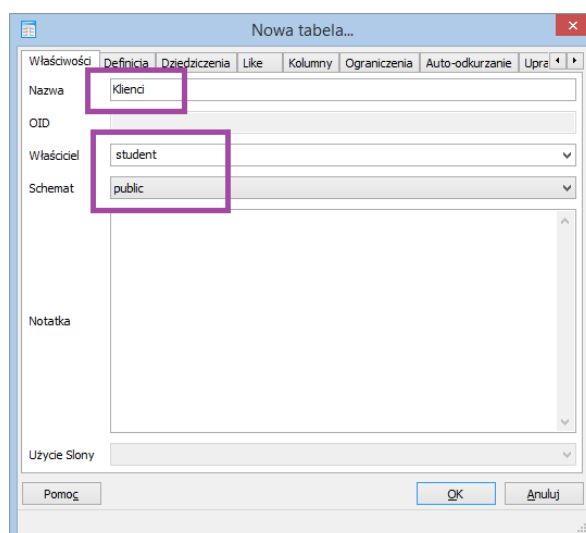
utworzonej bazy, jak także jej zawartość w postaci drzewiastej jest przedstawiona na poniższym rysunku.



- Należy utworzyć trzy tabele: *Klienci*, *Zamówienia*, *ZamówieniaKlienta*. Należy ustawić klucze główne i klucze obce.

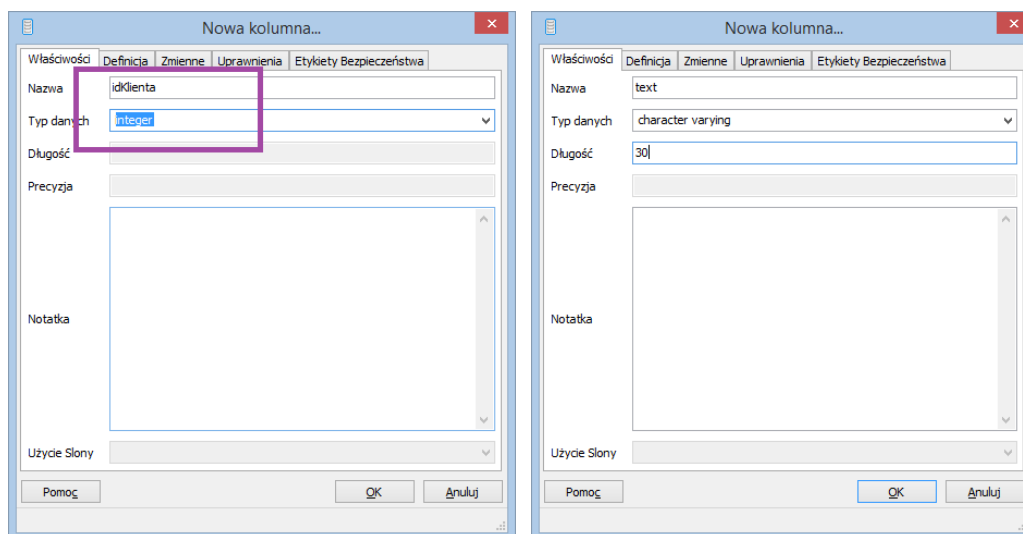
*Przykład tworzenia nowej tabeli Klienci jest przedstawiony poniżej.*

*Pierwsza zakładka wymaga podania nazwy nowej tabeli, jej właściciela oraz schematu, do którego będzie należała.*

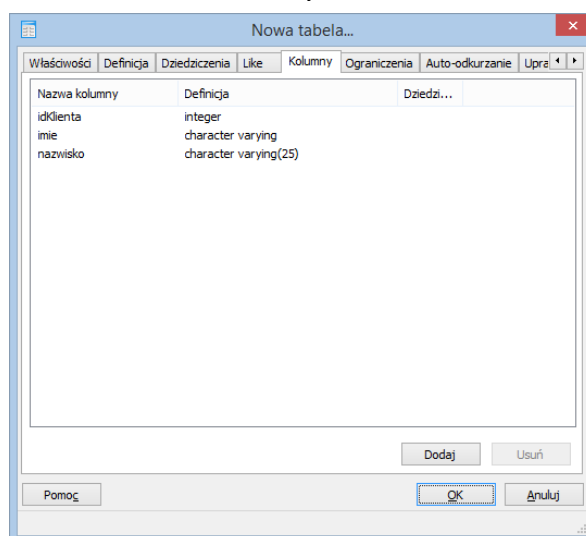


*W piątej zakładce definiowane są kolumny tabeli. Po kliknięciu przycisku Dodaj otwiera się nowe okno, w której należy podać nazwę i typ tworzonej kolumny, co przedstawiono poniżej.*

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



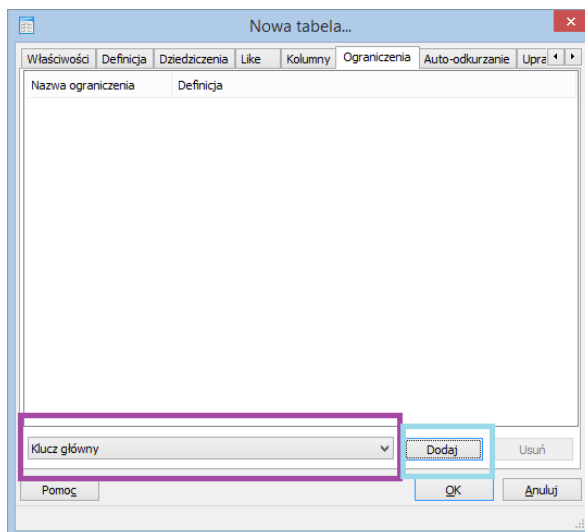
Po zatwierdzeniu parametrów nowej kolumny, kreator powraca do okna tworzenia tabeli, a także wyświetla wszystkie utworzone kolumny.



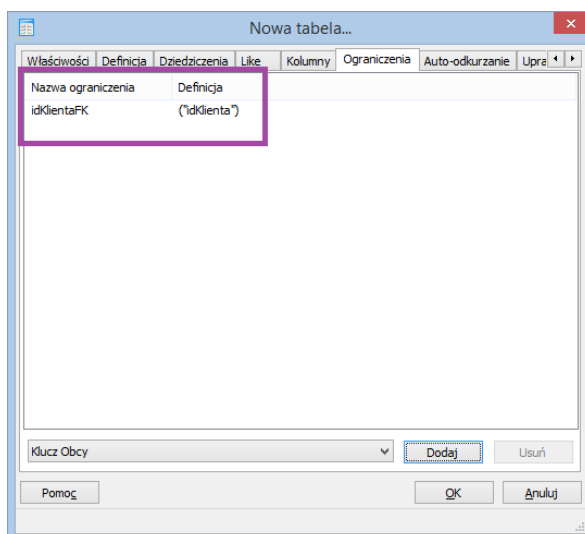
Nazwa kolumny	Definicja	Dziedz...
idklienta	integer	
imie	character varying	
nazwisko	character varying(25)	

Kolejnym krokiem, który powinien być wykonany jest przypisanie ograniczeń do tabeli. Dla tej tabeli zostanie zdefiniowany klucz główny, co zostało pokazane poniżej. W zakładce należy wybrać rodzaj ograniczenia (domyślnie, dla pierwszego ograniczenia podany jest klucz główny), a następnie kliknąć przycisk Dodaj.

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



*Po podaniu nazwy klucza głównego, wybraniu nazwy kolumny należy zatwierdzić wszystkie informacje.*



*Dla klucza obcego należy zdefiniować nazwę oraz wybrać kolumny należące do danych tabel. Po zatwierdzeniu dodawania nowej tabeli, zostanie ona utworzona i dodana do wybranego schematu bazy danych.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

4. Należy utworzyć tabele, zgodnie z ustalonym diagramem ERD na pierwszych laboratoriach przy użyciu skryptu SQL.

*Należy napisać skrypt SQL tworzący tabele i uruchomić go w edytorze SQL.*

*Przykład tworzenia tabeli Kursanci został przedstawiony poniżej. Należy pamiętać, żeby podać schemat, w którym ma zostać utworzona tabela.*

*Po wykonaniu zapytania, należy powrócić do bazy danych i odświeżyć listę tabel.*

```
CREATE SCHEMA IF NOT EXISTS laboratoriaSchema ;
```

```
CREATE TABLE IF NOT EXISTS laboratoriaSchema.Kursanci (  
    idKursanci INTEGER NOT NULL,  
    nazwisko VARCHAR(45) NOT NULL,  
    imie VARCHAR(45) NOT NULL,  
    drugieImie VARCHAR(45) NULL,  
    wojwodztwo VARCHAR(45) NOT NULL,  
    miasto VARCHAR(45) NOT NULL,  
    ulica VARCHAR(45) NOT NULL,  
    nr VARCHAR(45) NOT NULL,  
    nrMieszkania VARCHAR(45) NULL,  
    kodPocztowy VARCHAR(5) NULL,  
    email VARCHAR(45) NOT NULL,  
    tel VARCHAR(13) NULL,  
    pesel VARCHAR(11) NOT NULL,  
    PRIMARY KEY (idKursanci)  
);
```

## Inżynieria baz danych

### Laboratorium 6

Wstawianie danych do tabel bazy danych w systemie PostgreSQL

#### Cel zajęć

Umiejętność obsługi środowiska pgAdmin

Umiejętność wprowadzania rekordu do wybranej tabeli

Umiejętność wprowadzania danych do tabel przy użyciu skryptu SQL

#### Wymagania wstępne

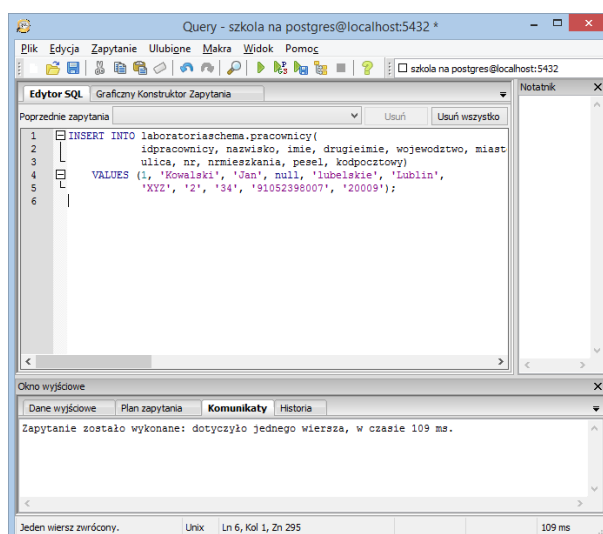
Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami encji.

Student posiada teoretyczną wiedzę dotyczącą środowiska pgAdmin i zna jego możliwości.

1. Należy wypełnić danymi utworzone w poprzednim laboratorium tabele.

*Należy wprowadzić dane po 1 rekordzie. W tym celu klikając prawym przyciskiem myszy na wybranej tabeli należy wybrać SKRYPTY->SKRYPT CREATE.*

*Poniższy przykład przedstawia wprowadzenie danych przy pomocy polecenia INSERT INTO dla tabeli Pracownicy.*



*Poniżej widnieje informacja, czy zapytanie zostało wykonane i jeśli tak to w jakim czasie.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Można teraz wyświetlić zawartość tabeli, przykładowo poprzez wybranie widoku tabeli dla zaznaczonej tabeli. Poniższy rysunek prezentuje wyświetlenie danych.*

	idpracownicy [PK] integer	nazwisko character var	imie character var	drugieimie character var	województwo character var	miasto character var	ulica character var
1	1	Kowalski	Jan		lubelskie	Lublin	XYZ
*							

*Ten sam efekt można uzyskać wybierając z rozwijanego menu opcji Przeglądanie danych. Można także wyświetlić dane poprzez polecenie **SELECT** w edytorze SQL lub skorzystać z graficznego narzędzia tworzenia zapytań.*

```
select * from laboratoriaschema.pracownicy;
```

2. Należy utworzyć skrypt SQL, który wypełni danymi wszystkie utworzone tabele.

## Inżynieria baz danych

### Laboratorium 7

Zarządzanie bazą danych PostgreSQL.

#### Cel zajęć

Umiejętność obsługi środowiska pgAdmin

Umiejętność tworzenia nowych użytkowników

Umiejętność tworzenia eksportu oraz importu bazy danych

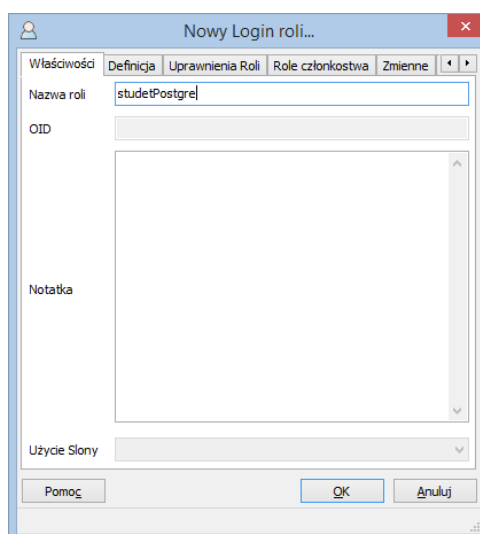
Umiejętność wykonywania kopii zapasowej

#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą środowiska pgAdmin i zna jego możliwości.

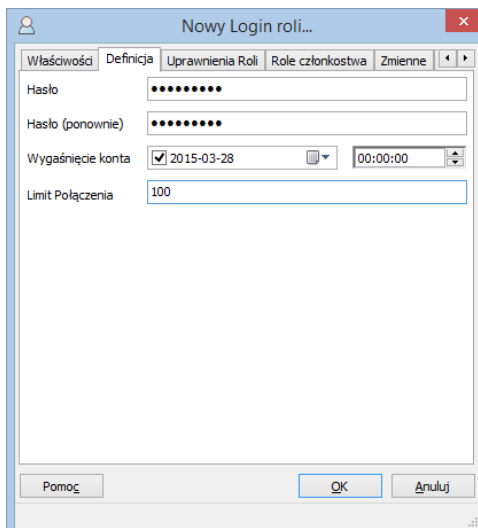
1. Należy utworzyć nową rolę o nazwie studentPostgre. Należy przypisać mu hasło oraz ustawić datę wygaśnięcia konta oraz limit połączenia na 100. Należy przydzielić mu uprawnienia super użytkownika.

*Należy z menu wybrać utworzenie nowej roli. W otworzonym kreatorze, w pierwszej zakładce należy zdefiniować nazwę użytkownika.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*W kolejnej zakładce należy podać hasło dla użytkownika, datę wygaśnięcia hasła oraz limit dla liczby połączeń.*



Nowy Login roli...

Właściwości Definicja Uprawnienia Roli Role członkostwa Zmienne

Hasło: .....

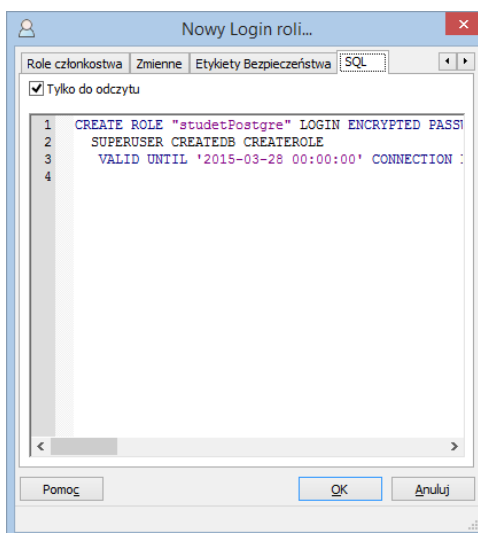
Hasło (ponownie): .....

Wygaśnięcie konta: ☒ 2015-03-28 00:00:00

Limit Połączenia: 100

Pomoc OK Anuluj

*W ostatniej zakładce można zobaczyć kod SQL, wygenerowany na podstawie podanych informacji.*



Nowy Login roli...

Role członkostwa Zmienne Etykiety Bezpieczeństwa SQL

☒ Tylko do odczytu

```

1 CREATE ROLE "studetPostgre" LOGIN ENCRYPTED PASSW
2 SUPERUSER CREATEDB CREATEROLE
3 VALID UNTIL '2015-03-28 00:00:00' CONNECTION :
4

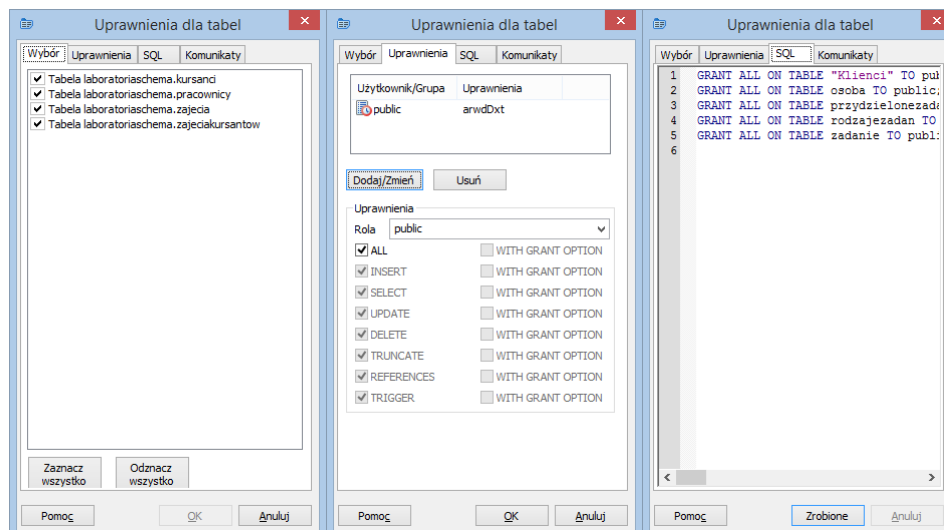
```

Pomoc OK Anuluj

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

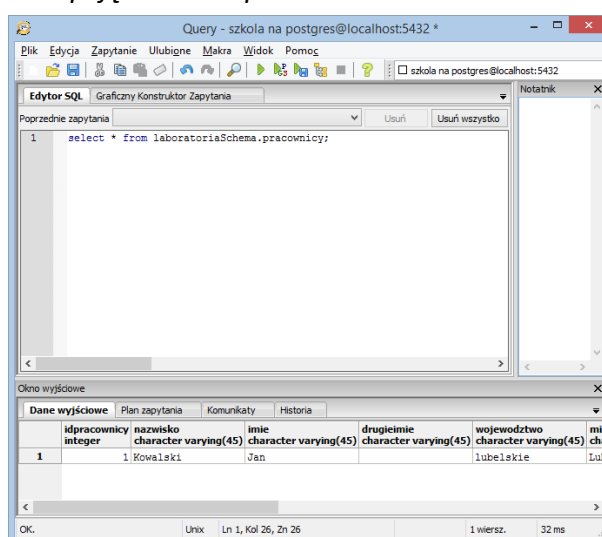
## 2. Należy nadać uprawnienia do stworzonych tabel

*W tym celu należy dla tabel wybrać uprawnienia dla tabel.*

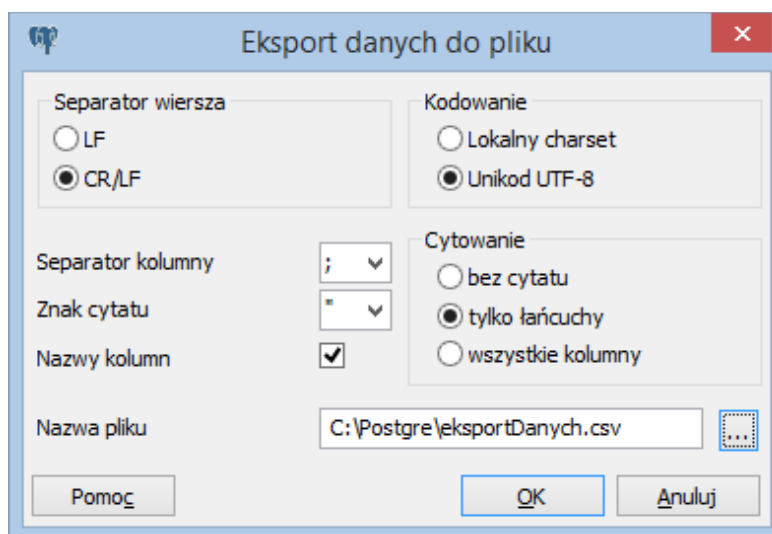


## 3. Należy wykonać eksport zawartość wszystkich danych wybranej tabeli do pliku przy użyciu narzędzia pgAdmin.

*Należy uruchomić edytor zapytań, wykonać zapytanie wybierające wszystkie dane z wybranej tabeli, a następnie wybrać opcję Plik->Eksport.*

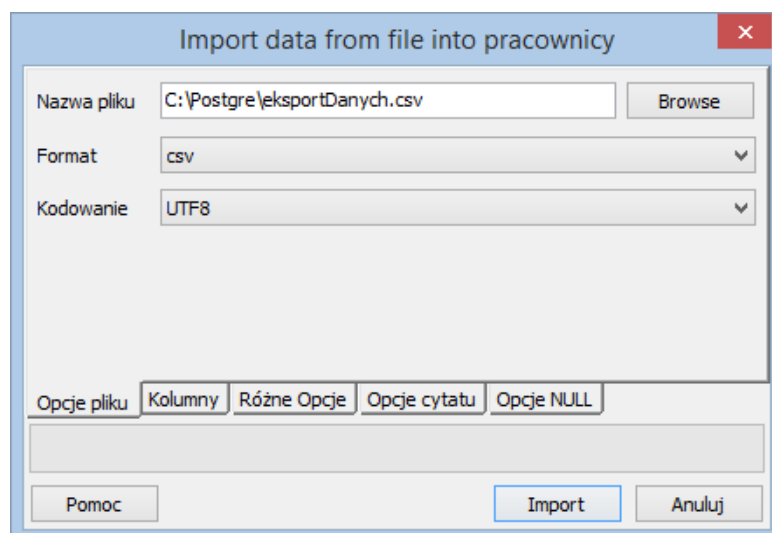


Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



4. Należy usunąć dane z tabeli, której dane zostały wyeksportowane.
5. Należy zaimportować dane z pliku do tabeli.

*W tym celu na wybranej tabeli z menu należy wybrać zaimportuj, a następnie wybrać opcje odpowiednie dla importu, co przedstawiono poniżej.*





Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

The dialog box is titled "Import data from file into pracownicy". It has a close button (X) in the top right corner. The main area contains three settings: "OID" with an unchecked checkbox, "Nagłówek" with a checked checkbox, and "Ogranicznik" with a dropdown menu showing a semicolon (;). Below these settings are five tabs: "Opcje pliku", "Kolumny", "Różne Opcje" (which is selected and highlighted with a dotted border), "Opcje cytatu", and "Opcje NULL". At the bottom of the dialog are three buttons: "Pomoc", "Done", and "Anuluj".

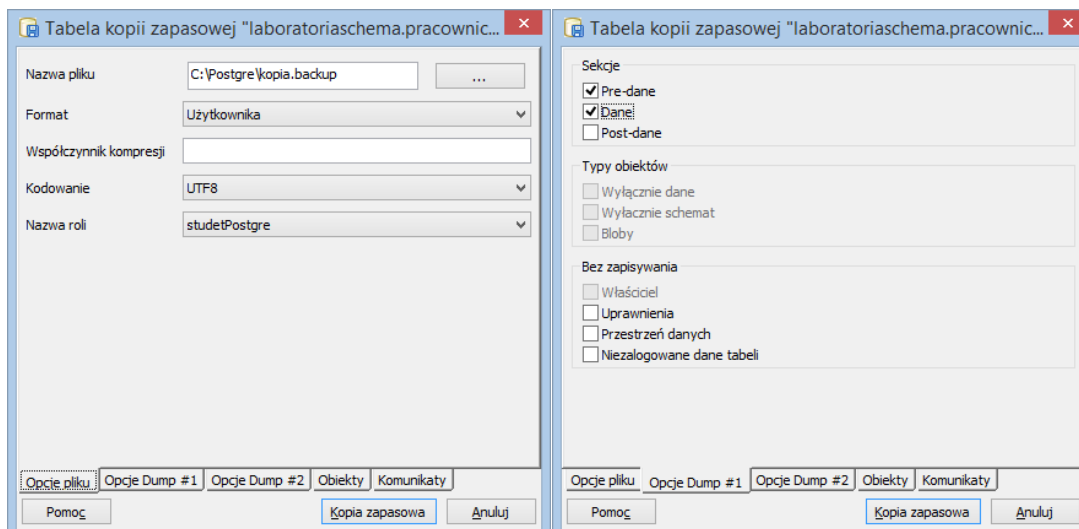
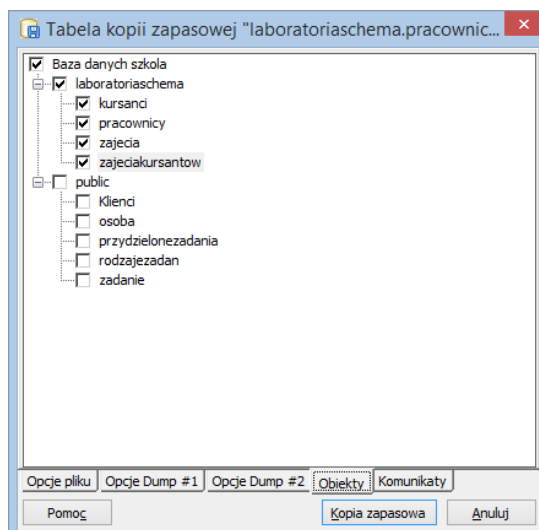
The dialog box is titled "Import data from file into pracownicy". It has a close button (X) in the top right corner. The main area contains two sections: "Ciąg znaków NULL" with an empty text input field, and "Zignorować dla kolumn" with a list of checkboxes. The list includes: "idpracownicy", "nazwisko", "imie", "drugieimie" (which is checked), "województwo", "miasto", "ulica", "nr", and "nrmieszkania". Below these sections are five tabs: "Opcje pliku", "Kolumny", "Różne Opcje", "Opcje cytatu", and "Opcje NULL" (which is selected and highlighted with a dotted border). At the bottom of the dialog are three buttons: "Pomoc", "Done", and "Anuluj".

6. Należy wykonać kopię zapasową bazy danych dla schematu labioratoriumSchema.

*Należy postępować zgodnie z instrukcjami oraz podać niezbędne informacje. Poniżej zamieszczono wzór.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

7. Należy usunąć schemat laboratoriaSchema.

*Do tego celu można użyć polecenia usuwania kaskadowego.*

8. Należy przywrócić schemat laboratoriaSchema.

*Należy uruchomić kreator przywracania danych (Narzędzia->Przywróć) .*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Przywrócenie bazy danych "szkola"

Format: Niestandardowy lub tar

Nazwa pliku: C:\Postgre\kopia.backup

Liczba zadań:

Nazwa roli: studetPostgre

Opcje pliku | Opcje Przywracania #1 | Opcje Przywracania #2 | Obiekty | Komu

Pomoc Wyświetl obiekty Przywróć Anuluj

Przywrócenie bazy danych "szkola"

Sekcje

- ☒ Pre-dane
- ☒ Dane
- ☒ Post-dane

Typy obiektów

- ☐ Wyłącznie dane
- ☐ Wyłącznie schemat

Bez zapisywania

- ☐ Właściciel
- ☐ Uprawnienia
- ☐ Przestrzeń danych

Opcje pliku | Opcje Przywracania #1 | Opcje Przywracania #2 | Obiekty | Komu

Pomoc Wyświetl obiekty Przywróć Anuluj

Przywrócenie bazy danych "szkola"

Zapytania

- ☐ Zawrzeć instrukcję CREATE DATABASE
- ☐ Wyczyść przed odtworzeniem
- ☐ Jedna transakcja

Wyłączony

- ☐ Wyzwalacz
- ☐ Brak Danych dla Nieudanych Tabel

Pozostałe / Zachowania

- ☐ Użyj SET SESSION AUTHORIZATION
- ☐ Wyjście jeśli błąd
- ☒ Szczegółowe komunikaty

Opcje pliku | Opcje Przywracania #1 | Opcje Przywracania #2 | Obiekty | Komu

Pomoc Wyświetl obiekty Przywróć Anuluj

## Inżynieria baz danych

### Laboratorium 8

Tworzenie bazy danych w systemie MS SQL

#### Cel zajęć

Umiejętność obsługi środowiska MS SQL Management Studio 2012

Umiejętność utworzenia bazy danych w narzędziu MS SQL Management Studio 2012

Umiejętność utworzenia tabel, ręcznie i z użyciem skryptu SQL

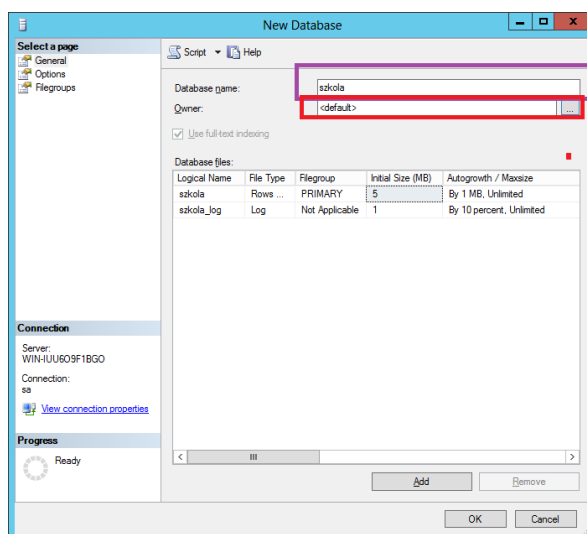
#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami encji.

Student posiada teoretyczną wiedzę dotyczącą środowiska MS SQL Management Studio 2012 i zna jego możliwości.

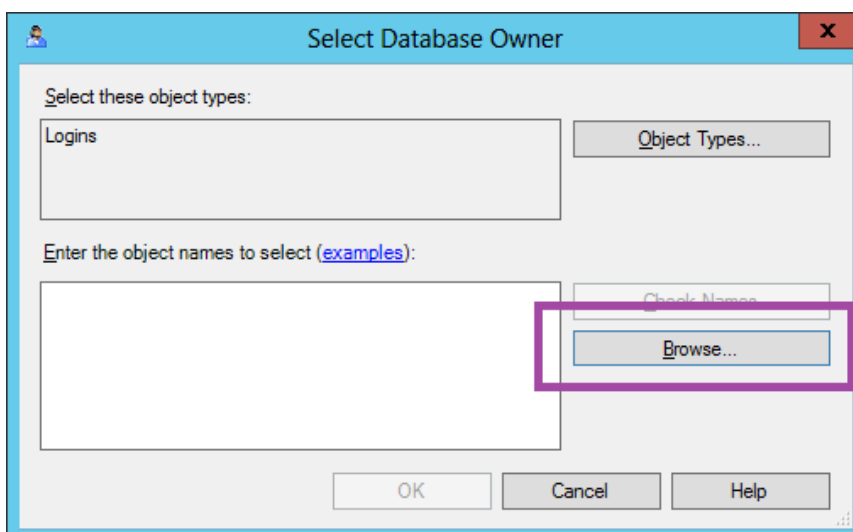
1. Należy utworzyć nową bazę danych o nazwie szkoła.

*Należy uruchomić środowisko MS SQL Management Studio. Po lewej stronie, w strukturze drzewiastej należy wybrać katalog Databases, a następnie z menu rozwijanego po kliknięciu prawym przyciskiem myszy opcję NEW DATABASE. W otwartym oknie należy wpisać nazwę tworzonej bazy.*

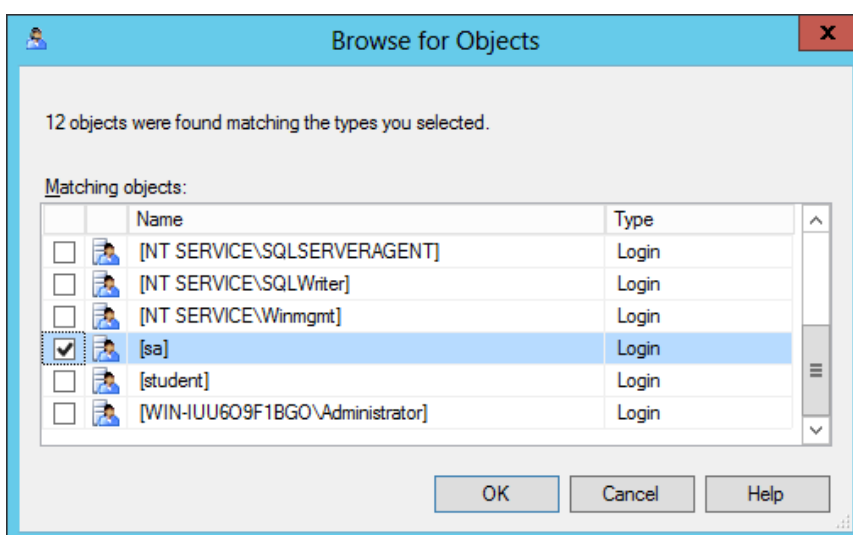


Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Należy przypisać właściciela do bazy danych. W tym celu trzeba rozwinąć listę użytkowników. Zostanie wyświetlone okno, które pozwala przefiltrować użytkowników – domyślnie jest ustawiony typ na login.*

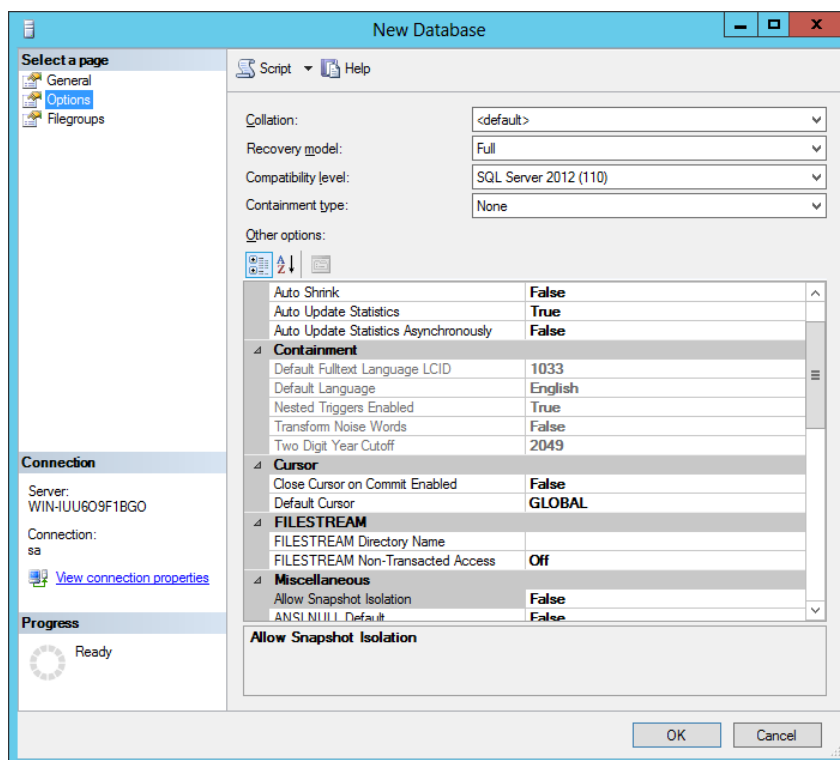


*Po wybraniu przycisku Browse zostanie otwarte kolejne okno, z użytkownikami. Należy wybrać jednego z nich i zatwierdzić.*



*Druga zakładka okna służącego do utworzenia bazy danych zawiera dodatkowe, opcjonalne informacje związane z bazą, które zostały przedstawione poniżej.*

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

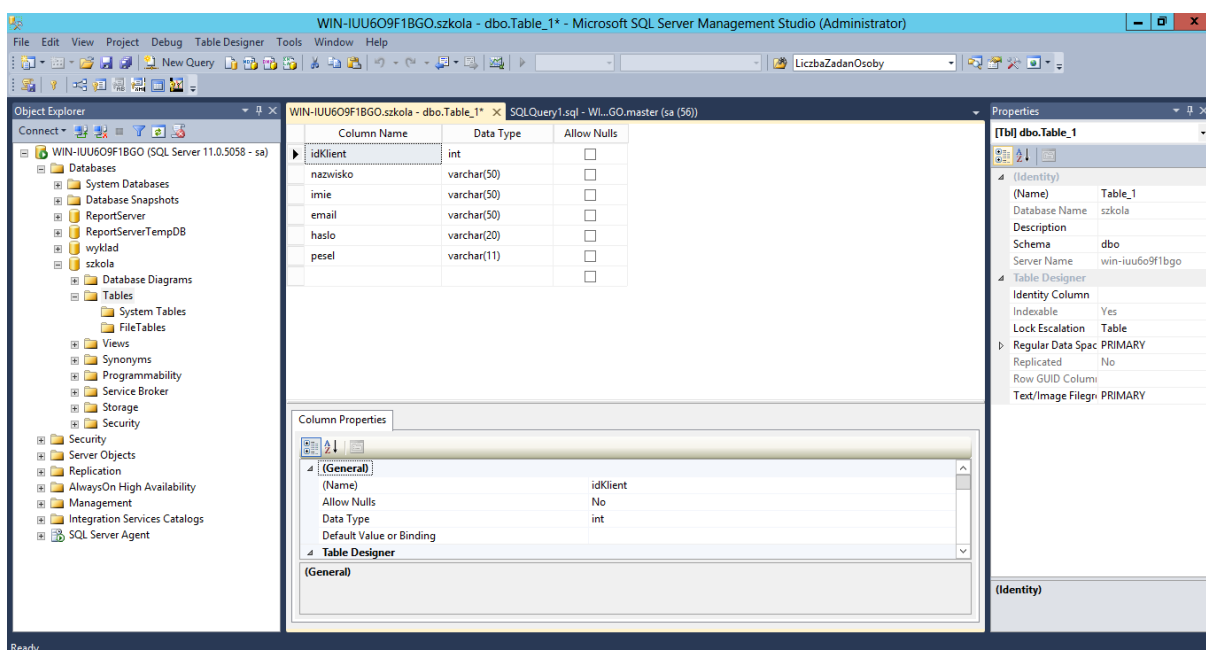


2. Należy utworzyć ręcznie trzy tabele w bazie danych: Klienci, Zamówienia, ZamówieniaKlienta.

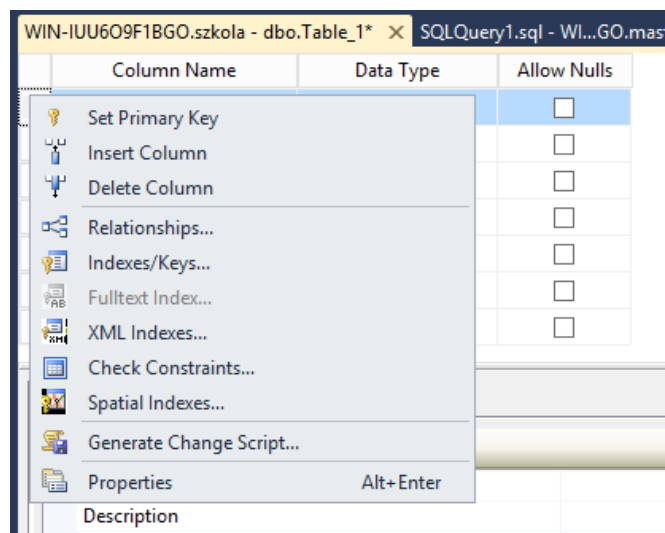
Aby utworzyć nową tabelę, należy z menu, pojawiającym się po kliknięciu prawym przyciskiem myszy na katalog Tables zjadającego się w strukturze drzewiastej bazy danych, wybrać opcję New Table. W oknie pojawi się możliwość wpisywania kolejnych kolumn tabeli. Dla każdej kolumny należy podać: nazwę, typ oraz czy kolumna przyjmuje wartości null. Na rysunku poniżej przedstawiono przykład tworzenia nowej kolumny Klienci.

W drugiej części, poniżej listy kolumn, znajdują się właściwości dla wybranej kolumny.

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



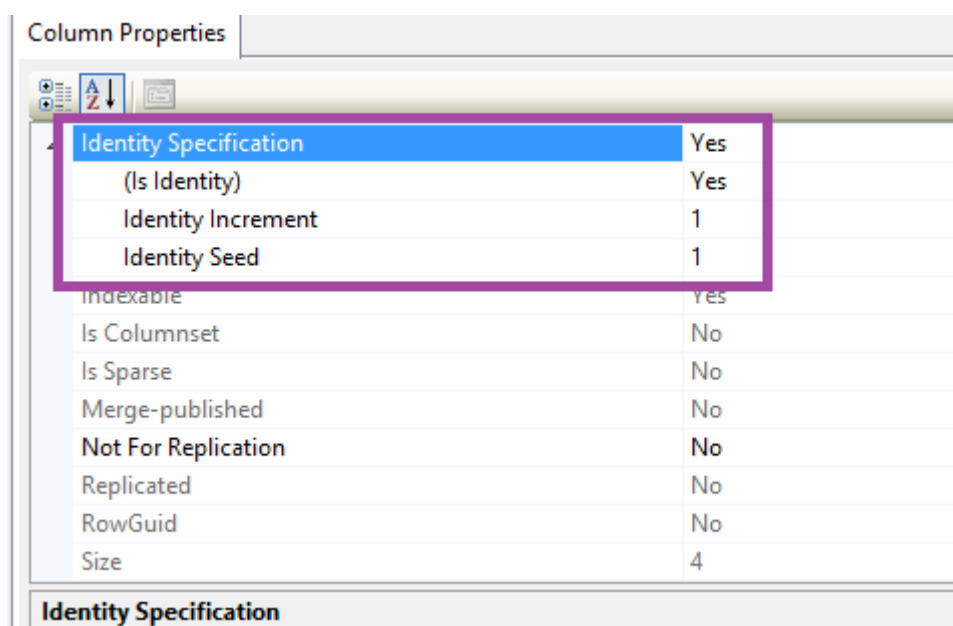
*Dla kolumny idKlient należy ustawić opcję klucza głównego. Dla zaznaczonej kolumny klikając prawym przyciskiem myszy, pojawia się menu, przedstawione poniżej. Należy wybrać 'Set Primary Key'*



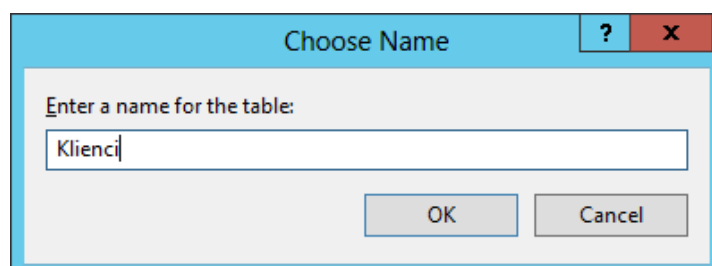
Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Ustawienie automatycznej inkrementacji wartości tabeli może być wskazana we właściwościach danej kolumny. Kolumna ta musi być typu całkowitego (int). Należy zaznaczyć opcje:*

- *Identity Specification na YES*
- *Is Identity na YES*
- *Identity Increment na 1*
- *Identity Seed 1*



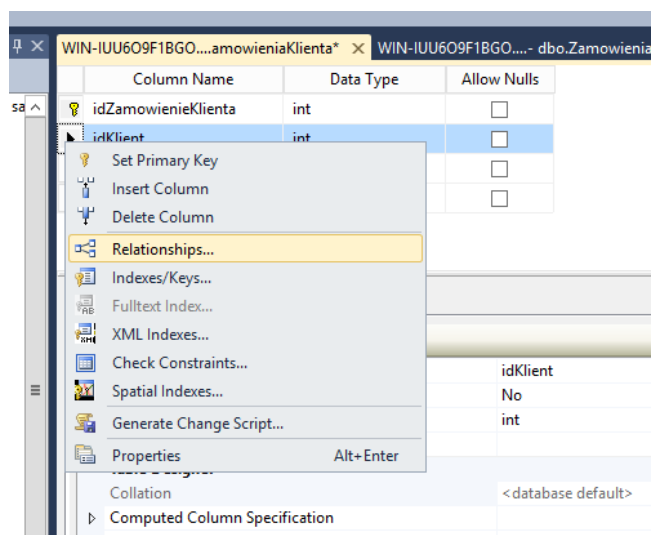
*Dopiero po zapisaniu wprowadzonych danych, pojawia się okno, w którym należy podać nazwę tworzonej tabeli. Przedstawiono to poniżej.*



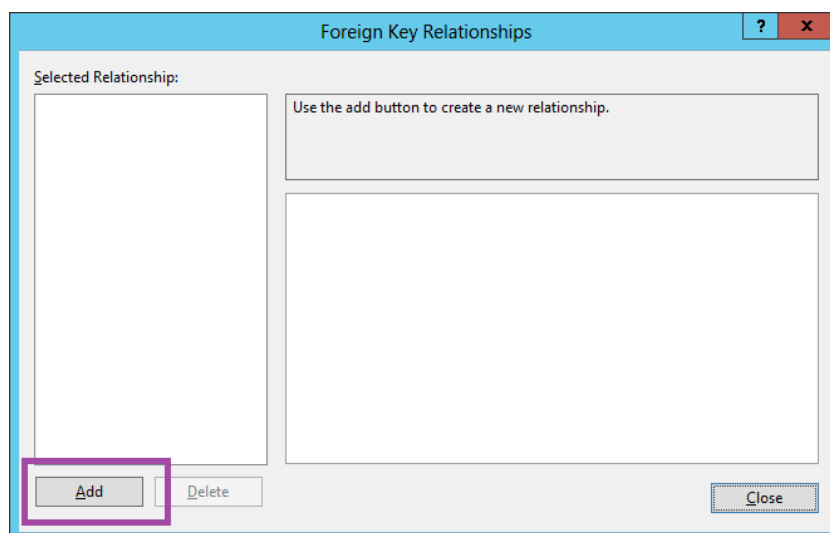
Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

3. Należy utworzyć relacje pomiędzy tabelami *Klienci*, *Zamówienia* oraz *ZamówieniaKlienci*.

*Prawym przyciskiem myszy na danej kolumnie tabeli i należy wybrać Relationship, co przedstawiono na poniższym rysunku.*



*Należy wybrać dodanie nowego związku:*



*Należy zdefiniować nazwę oraz wybrać kolumny i tabele do relacji.*





Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Tables and Columns

Relationship name:  
FK\_ZamowieniaKlienta\_Klienci

Primary key table:  
Klienci  
idKlient

Foreign key table:  
ZamowieniaKlienta  
idKlient

OK Cancel

4. Należy utworzyć nową tabelę o nazwie Pracownicy przy użyciu zapytania SQL.

*Należy wyświetlić treść zapytania tworzenia tabeli Klienci i na jej podstawie utworzyć nową tabelę.*

*Aby wyświetlić skrypt tworzący, należy na danej tabeli kliknąć prawym przyciskiem myszy, a następnie wybrać SCRIPT TABLE AS -> CREATE TO. Należy pozmieniać dane, a następnie wykonać skrypt.*

```
/****** Object: Table [dbo].[Klienci] Script Date: 2014-12-03 18:20:06 *****/
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Pracownicy](
    [idPracownik] [int] IDENTITY(1,1) NOT NULL,
    [nazwisko] [varchar](50) NOT NULL,
    [imie] [varchar](50) NOT NULL,
    [email] [varchar](50) NOT NULL,
    [haslo] [varchar](20) NOT NULL,
    [pesel] [varchar](11) NOT NULL,
    CONSTRAINT [PK_Pracownicy] PRIMARY KEY CLUSTERED
    (
        [idPracownik] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) GO

SET ANSI_PADDING OFF
GO
```

Messages  
Command(s) completed successfully.

Query executed successfully. WIN-IUU609F1BGO (11.0 SP2) sa (66) szkola 00:00:00 0 rows

5. Należy utworzyć tabele, zgodnie z ustalonym diagramem ERD na pierwszych zajęciach przy użyciu skryptu SQL.

## Inżynieria baz danych

### Laboratorium 9

Wstawianie danych do tabel bazy danych MS SQL

#### Cel zajęć

Umiejętność obsługi środowiska MS SQL Management Studio 2012

Umiejętność wprowadzania rekordu do wybranej tabeli

Umiejętność wprowadzania danych do tabel przy użyciu skryptu SQL

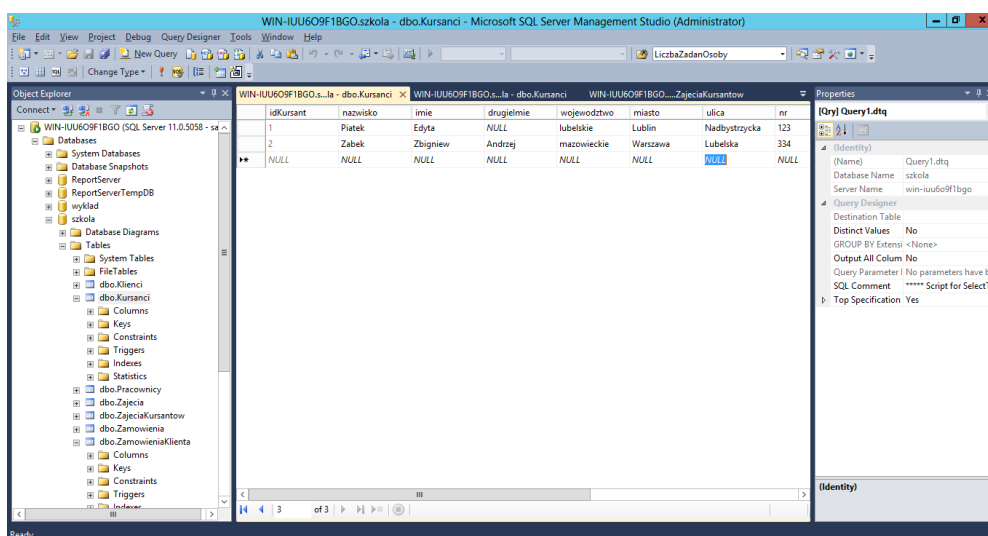
#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą modelowania struktury relacyjnej bazy danych, zna podstawowe pojęcia i rozumie różnice pomiędzy relacjami encji.

Student posiada teoretyczną wiedzę dotyczącą środowiska MS SQL Management Studio 2012i zna jego możliwości.

1. Należy wypełnić dwa rekordy tabeli Kursanci.

*W tym celu należy wybrać edytowanie pierwszych 200 (EDIT TOP 200 ROWS). Jeśli dla klucza głównego ustalono Identity, jego wartość jest ustalana automatycznie.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

2. Należy uzupełnić kilka rekordów tabeli pracownicy poprzez polecenie INSERT INTO.

*Należy uruchomić skrypt tworzenia nowego rekordu poprzez polecenie INSERT INTO.*

*Przykładowy skrypt dla tabeli Pracownicy został przedstawiony poniżej.*

```
USE [szkola]
GO
```

```
INSERT INTO [dbo].[Pracownicy]
    ([nazwisko]
    ,[imie]
    ,[drugieImie]
    ,[pesel]
    ,[województwo]
    ,[miasto]
    ,[ulica]
    ,[nr]
    ,[nrMieszkania]
    ,[dataZatrudnienia]
    ,[dataZwolnienia])
VALUES
    ('Kowalski'
    ,'Adrian'
    ,'Antoni'
    ,'87120387665'
    ,'lubelskie'
    ,'Zamość'
    ,'Kraśnicka'
    ,'56'
    ,null
    ,'2012-12-12'
    ,null)
```

```
GO
```

3. Należy uzupełnić pozostałe tabele z użyciem skryptu.
4. Należy utworzyć diagram utworzonej bazy danych.

## Inżynieria baz danych

### Laboratorium 10

Zarządzanie bazą danych MS SQL.

#### Cel zajęć

Umiejętność obsługi środowiska MS SQL Management Studio 2012

Umiejętność tworzenia nowych użytkowników

Umiejętność tworzenia eksportu oraz importu bazy danych

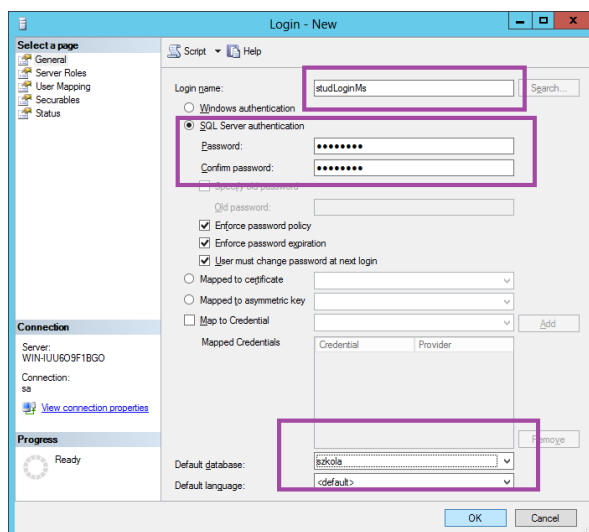
Umiejętność wykonywania kopii zapasowej

#### Wymagania wstępne

Student posiada teoretyczną wiedzę dotyczącą środowiska MS SQL Management Studio 2012 i zna jego możliwości.

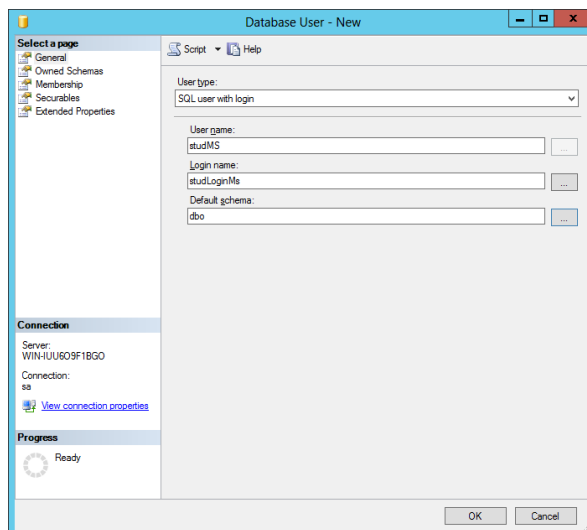
1. Należy utworzyć nowy login o nazwie studLoginMS.

Należy wybrać *Security->Logins->New Login*. Tworzenie nowego loginu prezentuje poniższy rysunek.



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

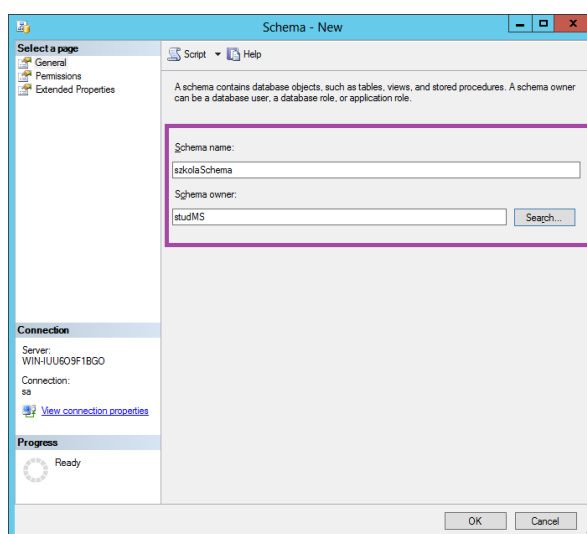
## 2. Utwórz użytkownika o nazwie studMS na podstawie utworzonego loginu.



## 3. Utwórz innego użytkownika przy użyciu polecenia CREATE USER.

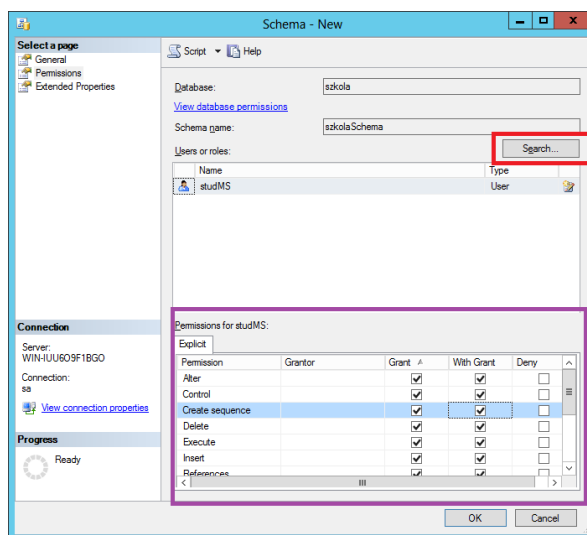
## 4. Utwórz nowy schemat o nazwie szkolaSchema.

*Poniższe rysunki przedstawiają kolejne kroki tworzenia nowego schematu. Najpierw podawana jest nowa nazwa i wybierany jest właściciel schematu wśród użytkowników bazy danych.*



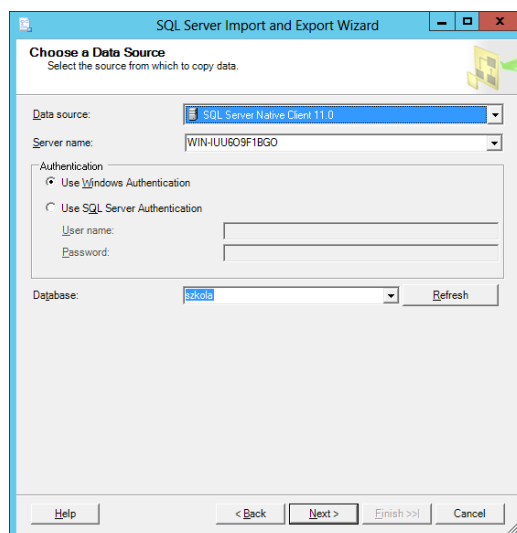
Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Należy przydzielić mu użytkownika studMS, a także uprawnienia.*



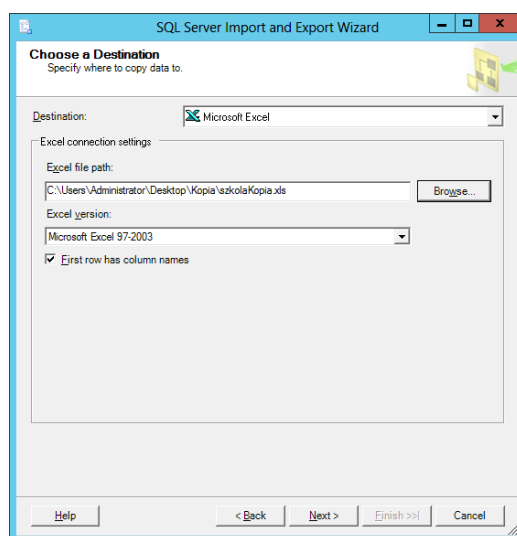
5. Wykonaj eksport bazy danych o nazwie *szkola*.

*Należy uruchomić narzędzie eksportu bazy danych (Task->Export data).*

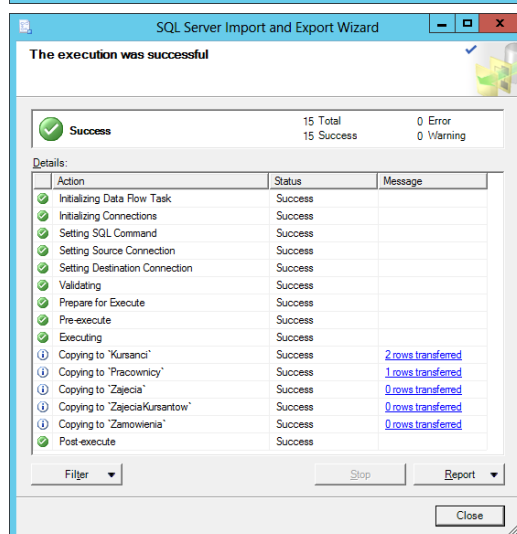
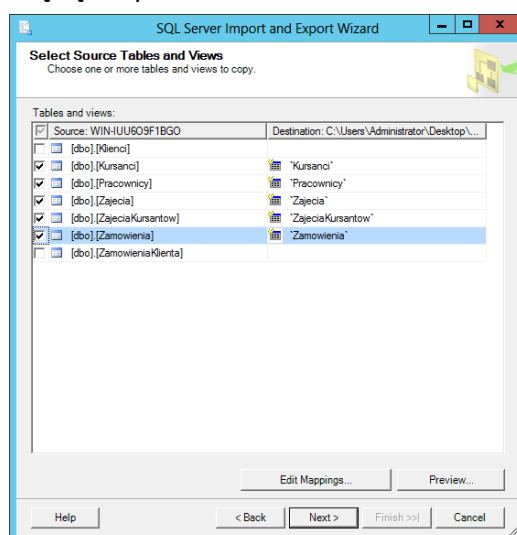


*Należy wybrać do jakiego rodzaju pliku zostaną wyeksportowane.*

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego



*Należy wybrać które tabele będą eksportowane.*

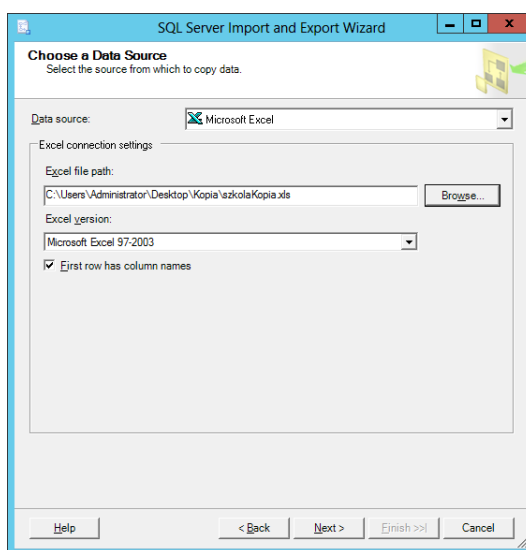


Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

## 6. Usuń wyeksportowane tabele.

*Uwaga! Jeśli tabela jest powiązana z inną, trzeba usunąć tę w związku znajdującą się postronnie wiele.*

7. Przeprowadź import danych z pliku, do którego wyeksportowano dane w zadaniu 5. Należy pamiętać, aby w pierwszej kolejności podać źródło danych, a następnie bazę.



SQL Server Import and Export Wizard

**Choose a Data Source**  
Select the source from which to copy data.

Data source: **Microsoft Excel**

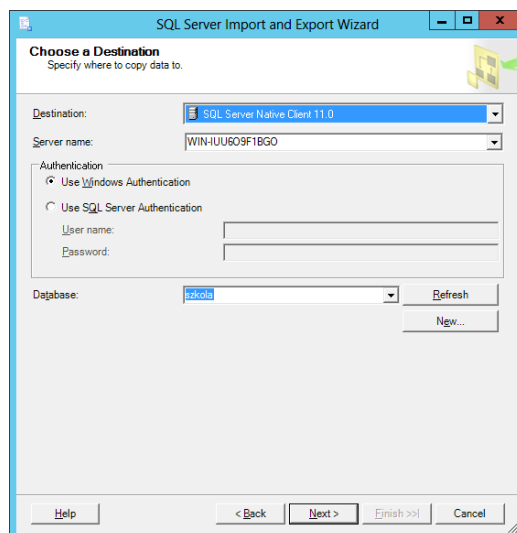
Excel connection settings

Excel file path: C:\Users\Administrator\Desktop\Kopia\ezkolaKopia.xls **Browse...**

Excel version: Microsoft Excel 97-2003

☒ First row has column names

**Help** < Back Next > Finish >> Cancel



SQL Server Import and Export Wizard

**Choose a Destination**  
Specify where to copy data to.

Destination: **SQL Server Native Client 11.0**

Server name: WIN-IUU609F1BGO

Authentication

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

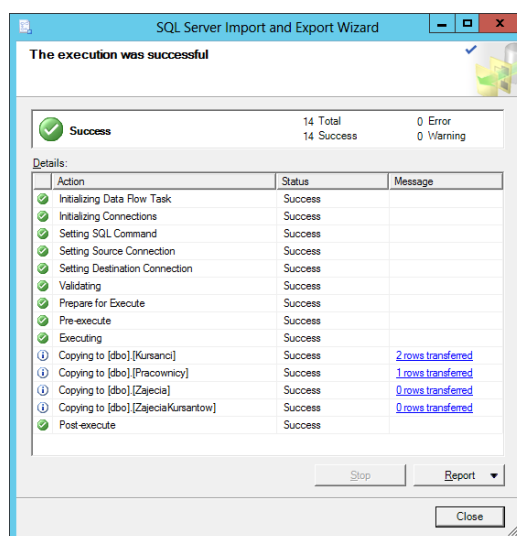
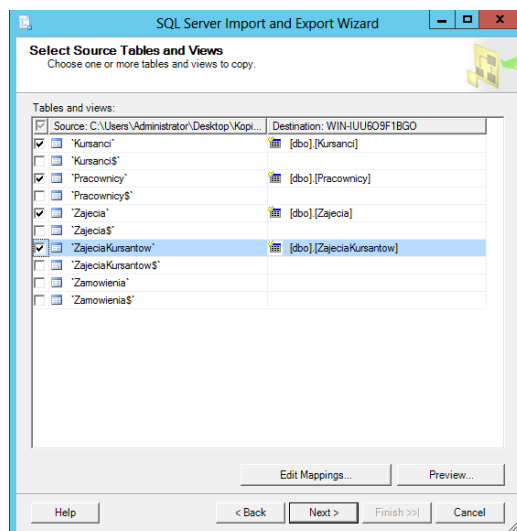
Database: ezkola **Refresh** **New...**

**Help** < Back Next > Finish >> Cancel



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

*Następnie należy wybrać kolumny do eksportu danych.*



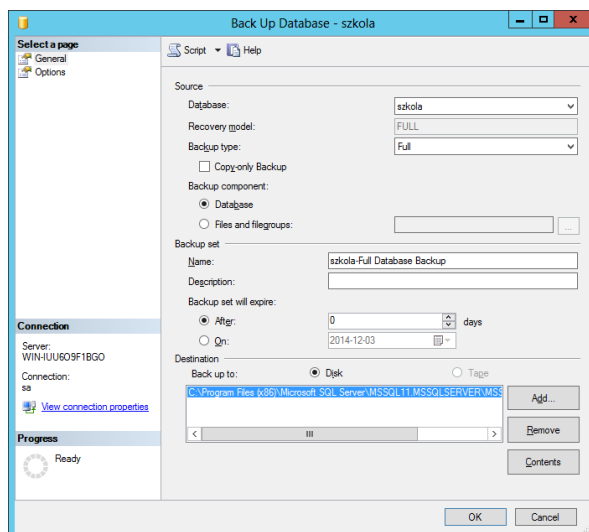
*Po odświeżeniu listy tabel w bazie danych, powinny pojawić się zaimportowane tabele z danymi.*



Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

## 8. Wykonaj kopię bazy danych.

*Dla wybranej bazy należy wybrać Task->Back up.*



## Inżynieria baz danych

### Laboratorium 11

Programowanie w języku T-SQL.

#### Cel zajęć

---

Umiejętność tworzenia prostych programów w języku T-SQL

Umiejętność stosowania instrukcji warunkowych i iteracyjnych w języku T-SQL

Umiejętność tworzenia procedur i funkcji

Umiejętność tworzenia i posługiwania się kursorami

#### Wymagania wstępne

---

Student posiada teoretyczną wiedzę dotyczącą środowiska MS SQL Management Studio 2012. Student posiada umiejętność wykonywania zapytań SQL. Posiada on także podstawową wiedzę o programowaniu w języku T-SQL.

1. Napisz prosty program, który korzysta z jednej zmiennej typu tekstowego przechowującej informacje o imieniu. Należy przypisać jej wartość, a następnie wyświetlić.

*Przykład działania programu jest przedstawiony poniżej:*

```
DECLARE
    @imie varchar(30)
SET @imie = 'Ola'
PRINT 'Witaj ' + @imie + '!'
```

2. Napisz program sumę trzech liczb całkowitych i wyświetlający ten wynik w postaci: Suma trzech liczb: ..... wynosi ..... W miejscu kropek należy wstawić wartości trzech liczb oraz obliczony wynik.

```
DECLARE
    @a int,
    @b int,
    @c int,
    @w int
SET @a = 23
SET @b = 13
```

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

```
SET @c = 2
SET @w = @a + @b + @c;
PRINT 'Suma trzech liczb ' + convert(varchar(2), @a) + ' i ' +
convert(varchar(2), @b) + ' i ' + convert(varchar(2), @c) + ' wynosi ' +
convert(varchar(2), @w);
```

3. Napisz program obliczający wynik reszty z dzielenia dwóch liczb. Wynik powinien zostać wyświetlony w postaci słownej odpowiedzi.
4. Napisz program, który pobierze bieżącą datę, a następnie ją wyświetli.
5. Napisz program, który wyświetli nazwisko, imię oraz email wszystkich pracowników w postaci „nazwisko imię – email”.

```
DECLARE
@imie varchar(30),
@nazwisko varchar(30),
@email varchar(30)
SELECT @nazwisko = nazwisko, @imie = imie, @email = email
from dbo.Kursanci;
PRINT @nazwisko + ' - ' + @imie + ' - ' + @email
```

6. Napisz program, który wyświetli tych Kursantów, którzy posiadają identyfikator większy od wartości zmiennej podanej w programie.
7. Napisz program, który wyświetli dane tych Zajeć, które rozpoczynają się ciągu znaków podanego jako zmienna w programie.  
Podpowiedź: można skorzystać z operatora LIKE.
8. Napisz program wyświetlający wynik dzielenia dwóch liczb. Należy sprawdzić poprawność danych (instrukcją warunkową).

```
DECLARE
@dzielna int = 7,
@dzielnik int = 3,
@wynik float
IF @dzielnik != 0
BEGIN
SET @wynik = @dzielna / @dzielnik;
PRINT 'Wynik dzielenia liczb ' + convert(varchar(4), @dzielna) + ' ' +
convert(varchar(4), @dzielnik) + ' to ' + convert(varchar, @wynik)
END
ELSE
PRINT 'Dzielenie przez 0!'
```

9. Napisz program, który pobierze zmienną, a następnie wyświetli komunikat zgodnie z tabelką poniżej:

<i><b>Wartość</b></i>	<i><b>Tekst</b></i>
<5	Nic nie wygrałeś
[5; 10]	Wygrałeś kubek
[11,30]	Wygrałeś tablet
[31;60]	Wygrałeś laptop

10. Napisz program, który odczyta liczbę prowadzonych zajęć przez pracowników. Dany pracownik powinien być wybrany na bazie zmiennej – odnoszącej się do identyfikatora. Program powinien wyświetlić komunikat uzależniony od zwróconej liczby:
- Mało prowadzonych zajęć (od 0 do 2)
  - Sporo prowadzonych zajęć (od 3 do 8)
  - Dużo zajęć (powyżej 8)
11. Napisz program, który sprawdzi, czy odczytana liczba zajęć pracownika jest parzysta czy nieparzysta – należy zastosować instrukcję CASE.
12. Przerób program z zadania 10 w taki sposób, aby zastosować instrukcję CASE.
13. Napisz program (z użyciem instrukcji CASE), który wyświetli dane wszystkich pracowników i je posortuje. Jeśli dana osoba posiada identyfikator podzielny przez 2 to dane mają być posortowane rosnąco, w przeciwnym razie malejąco.  
Podpowiedź: można zastosować instrukcję WHEN polecenia CASE
14. Napisz program, który wyświetli dane zajęć i je posortuje. Należy posortować według godziny zajęć w dni tygodnia. Od poniedziałku do środy dane mają być posortowane malejąco, a w pozostałe rosnąco.
15. Napisz program wyświetlający kolejne liczby całkowite z zakresu od 1 do 20.  
Należy zastosować instrukcję iteracyjną (np. WHILE).
16. Napisz program, który odczyta identyfikator pracownika, który był ostatnio zwolniony, a następnie wyświetli dane pracowników, którzy posiadają identyfikator mniejszy. W przypadku, gdy zapytanie nie zwróci wyników należy wyświetlić odpowiedni komunikat.

### *Funkcje i procedury*

17. Napisz funkcję, która zwróci liczbę kursantów.

```
CREATE FUNCTION szkołaSchema.LiczbaKursantow()  
RETURNS int  
AS  
BEGIN  
    DECLARE @licz int  
    SET @licz = (select COUNT(*)  
    from dbo.Kursanci  
    )  
    RETURN (@licz)  
END;
```

18. Napisz funkcję, która zwróci maksymalną liczbę - który pracownik prowadzi najwięcej zajęć. Funkcja nie musi posiadać parametrów. Funkcja zwraca liczbę całkowitą.
19. Należy napisać program, który wywoła funkcję i wyświetli wynik zgodnie z instrukcją do zadania 10.
20. Należy napisać funkcję, która zliczy ile zajęć prowadzi dany pracownik. Funkcja powinna posiadać parametr w postaci identyfikatora pracownika. Funkcja powinna zwracać liczbę całkowitą.
21. Należy napisać program korzystający z funkcji z zadania 19.
22. Napisz funkcję, która wyświetli zajęcia prowadzone w danej filii. Nr filii podawany jest jako parametr funkcji.
23. Należy napisać program korzystający z utworzonej funkcji.
24. Zmodyfikuj wybraną funkcję poleceniem ALTER FUNCTION.
25. Napisz procedurę wyświetlającą dane pracowników zatrudnionych po roku 2010. Dane posortuj według daty zatrudnienia.
26. Uruchom utworzoną procedurę.
27. Napisz procedurę, która wyświetli liczbę prowadzonych zajęć przez pracowników o danym imieniu i nazwisku. Imię oraz nazwisko ma być parametrem procedury. Procedura powinna zwrócić dane przy użyciu parametrów.
28. Napisz program, który poda imię i nazwisko pracownika, a następnie wyświetli jego zajęcia z użyciem opracowanej procedury. Jeśli pracownik nie posiada prowadzonych zajęć, należy wyświetlić odpowiedni komunikat.
29. Należy napisać procedury wprowadzające dane do wszystkich kolejnych tabel utworzonych w bazie *szkola*. Każda procedura powinna dotyczyć konkretnej tabeli. Każda procedura powinna posiadać parametry – dane, które będą wprowadzone jako kolejne rekordy.

Publikacja/modyfikacja współfinansowana ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

30. Należy przetestować wszystkie opracowane procedury.

#### *Kursory*

31. Utwórz nowy kursor pobierający wszystkie dane o zajęciach, posortowanych według prowadzących.
32. Należy otworzyć kursor, przetworzyć kolejne rekordy (można je wyświetlić z odpowiednim formatowaniem) oraz zamknąć kursor.
33. Należy napisać program z użyciem kursora, który pobierze dane o wszystkich pracownikach. Program powinien wyświetlić, czy dany pracownik posiada status „zatrudniony” czy „zwolniony”.