



Uniwersytet Gdański
Wydział Matematyki, Fizyki i Informatyki
Instytut Informatyki

Schemat systemu informatycznego obsługującego system wirtualnej uczelni

Damian Pawłowski

Projekt z przedmiotu bazy danych na kierunku informatyka profil ogólnoakademicki na Uniwersytecie Gdańskim.

Gdańsk
24 maja 2020

Spis treści

1	Wprowadzenie	2
2	Opis projektu	2
2.1	Potencjalne grupy użytkowników	2
2.2	Wymagania funkcjonalne	3
2.3	Wymagania niefunkcjonalne	3
2.4	Diagram związków encji	3
3	Przykłady realizacji bazy danych	8
3.1	Przykłady zawartości najważniejszych tabel	8
3.2	Przykłady kilku zapytań i ich wyników	10

1 Wprowadzenie

Baza danych przeznaczona jest dla szkoły wyższej.

Ma ona na celu zmodernizowanie szkoły za pomocą elektronicznego sposobu przechowywania danych.

Ułatwi ona dokumentację najważniejszych informacji takich jak oceny czy wpłaty.

Umożliwi ona dostęp do internetowego planu lekcji dla studentów oraz nauczycieli.

Baza znajdować się będzie na zewnętrznym serwerze.

Baza jest znormalizowana do 3 Postaci Normalnej.

2 Opis projektu

Taka baza to idealne rozwiązanie dla nowo powstałej lub staroświeckiej uczelni.

Umożliwia ona zastąpienie tradycyjnego papierowego dziennika znacznie nowocześniejszym.

Pozwala ona na bezpieczne przechowywanie informacji o studentach , nauczycielach , ocenach , akademikach , wykładach oraz o pozostałych rzeczach związanych z funkcjonowaniem szkoły.

System umożliwia błyskawiczne liczenie średnich uczniów, porównywanie najlepszych ocen, szybkie sprawdzanie danych.

2.1 Potencjalne grupy użytkowników

- Administrator – Główny zarządca bazy danych, posiada pełen dostęp do bazy danych.
- Pracownicy uczelni – Mają dostęp do informacji o uczniach, ocenach, salach, planach lekcji.
- Studenci – Mają dostęp do swoich ocen, wpłat, świadczeń oraz informacji ogólnodostępnych o uczelni.
- Goście – Mają dostęp do informacji ogólnodostępnych np o kierunkach uczelni czy akademikach.

2.2 Wymagania funkcjonalne

Baza będzie przechowywać dane osobowe, adresy, oceny, cenniki, informacje o wydziałach, telefony, adresy email, pensje, daty. Jej zadania:

- tworzenie planów lekcji
- monitorowanie ocen
- wyświetlanie inf. o studentach i nauczycielach
- wyświetlanie inf. pracach dyplomowych
- wyświetlanie inf. akademikach
- dokumentowanie wpłat oraz świadczeń

2.3 Wymagania niefunkcjonalne

Baza znajduje się na zewnętrznym serwerze freemysqlhosting.net. Jego największą zaletą jest to, że jest za darmo. Darmowa wersja posiada pewne ograniczenie pamięci. Lecz zawsze można ulepszyć ją do wersji płatnej. Jeśli jest to nowa szkoła to takie rozwiązanie będzie najlepsze na kilka pierwszych miesięcy.

Do danych serwera można dostać się za pomocą phpmyadmin.

System zarządzania bazą danych to MySQL 5.0.

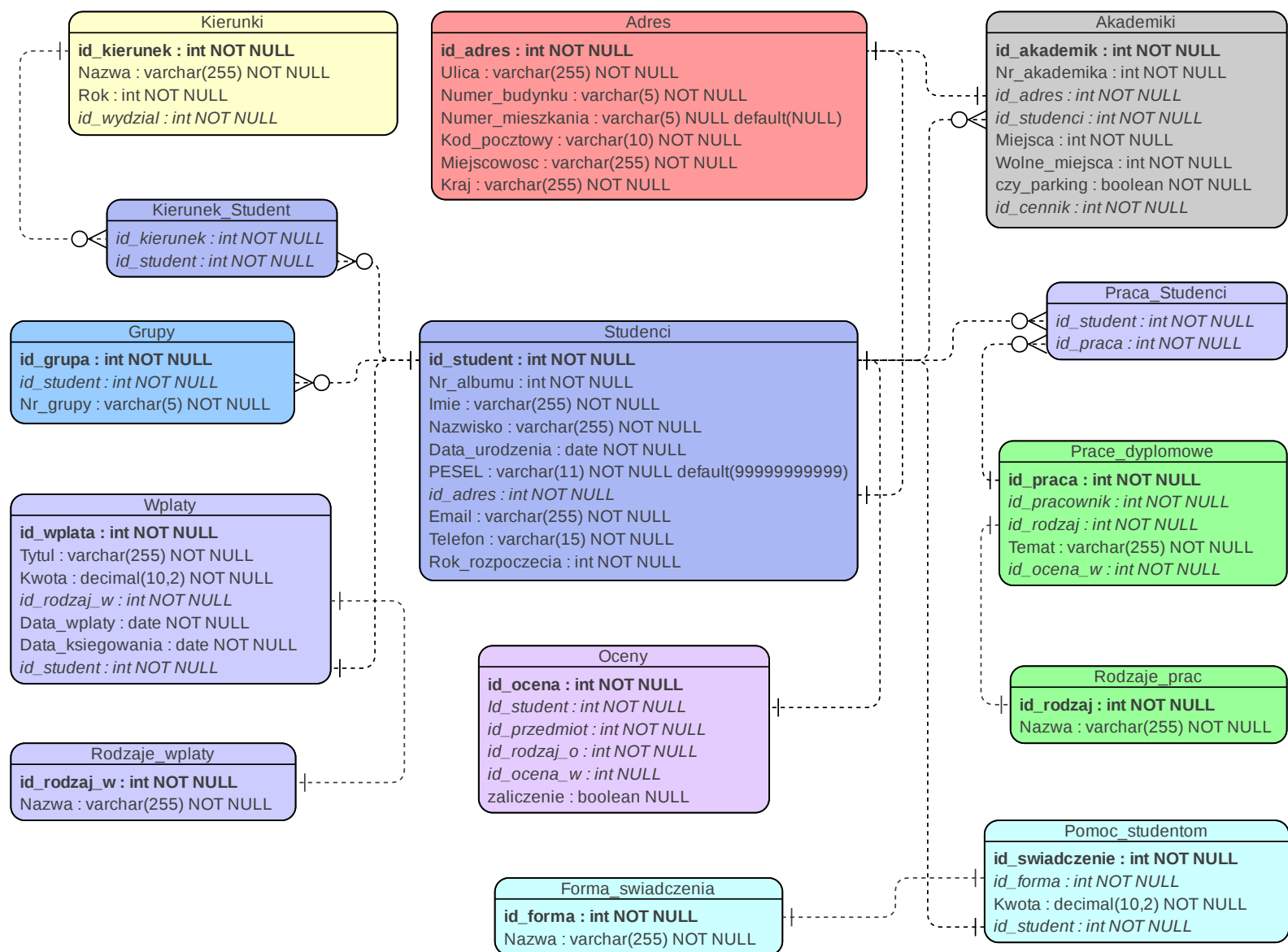
Do jego zalet należą:

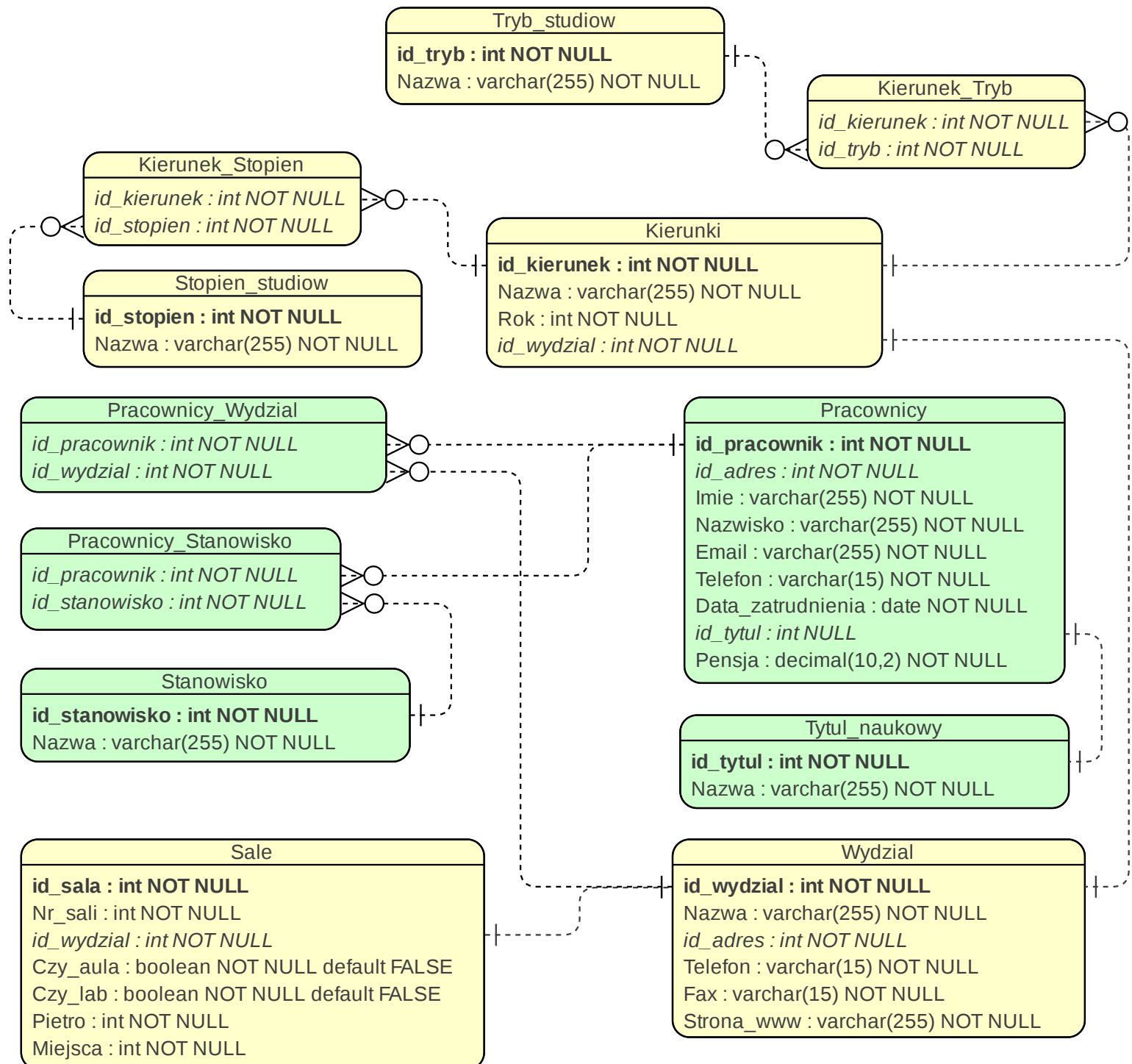
- Solidność
- Innowacyjność
- Łatwość w obsłudze
- Szybkość
- Open-Source

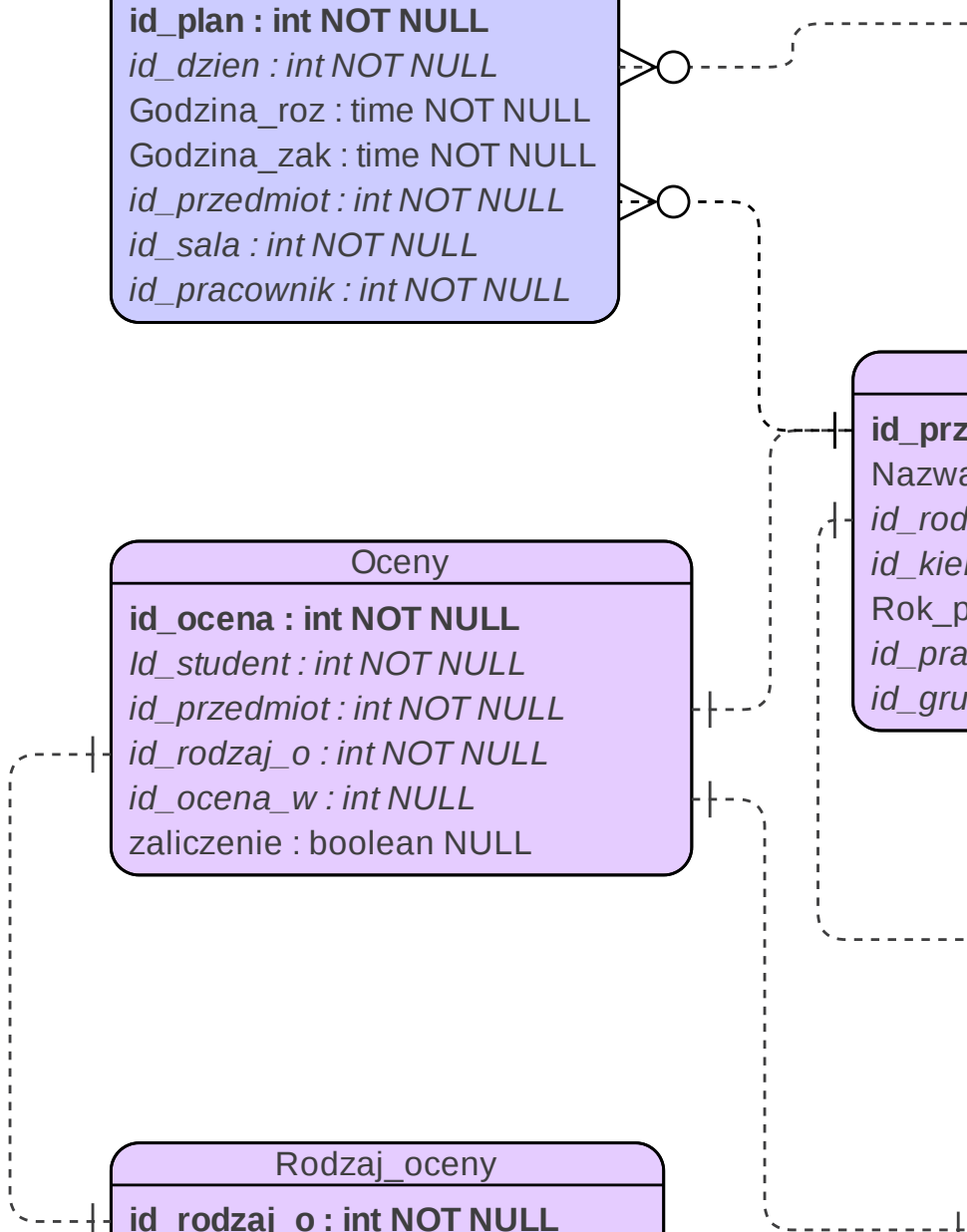
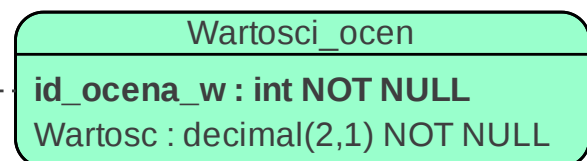
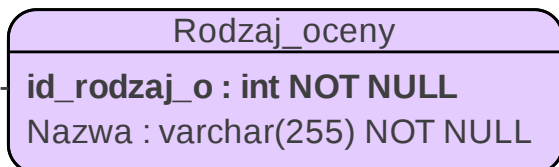
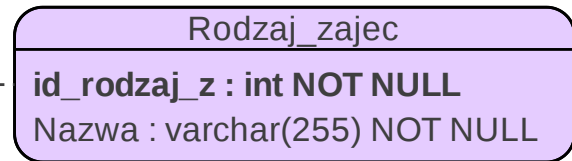
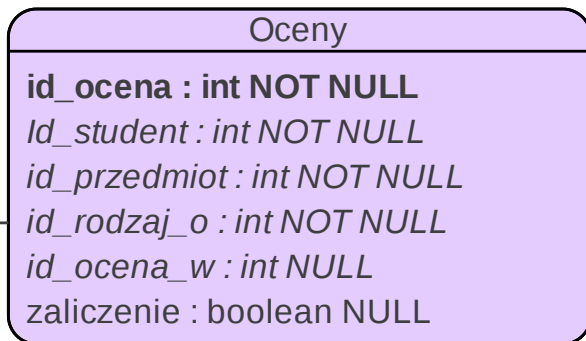
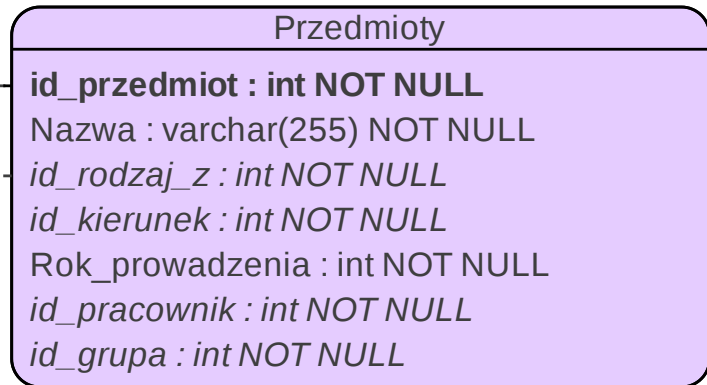
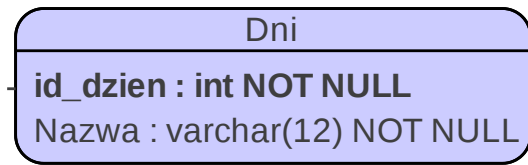
Do wad można zaliczyć:

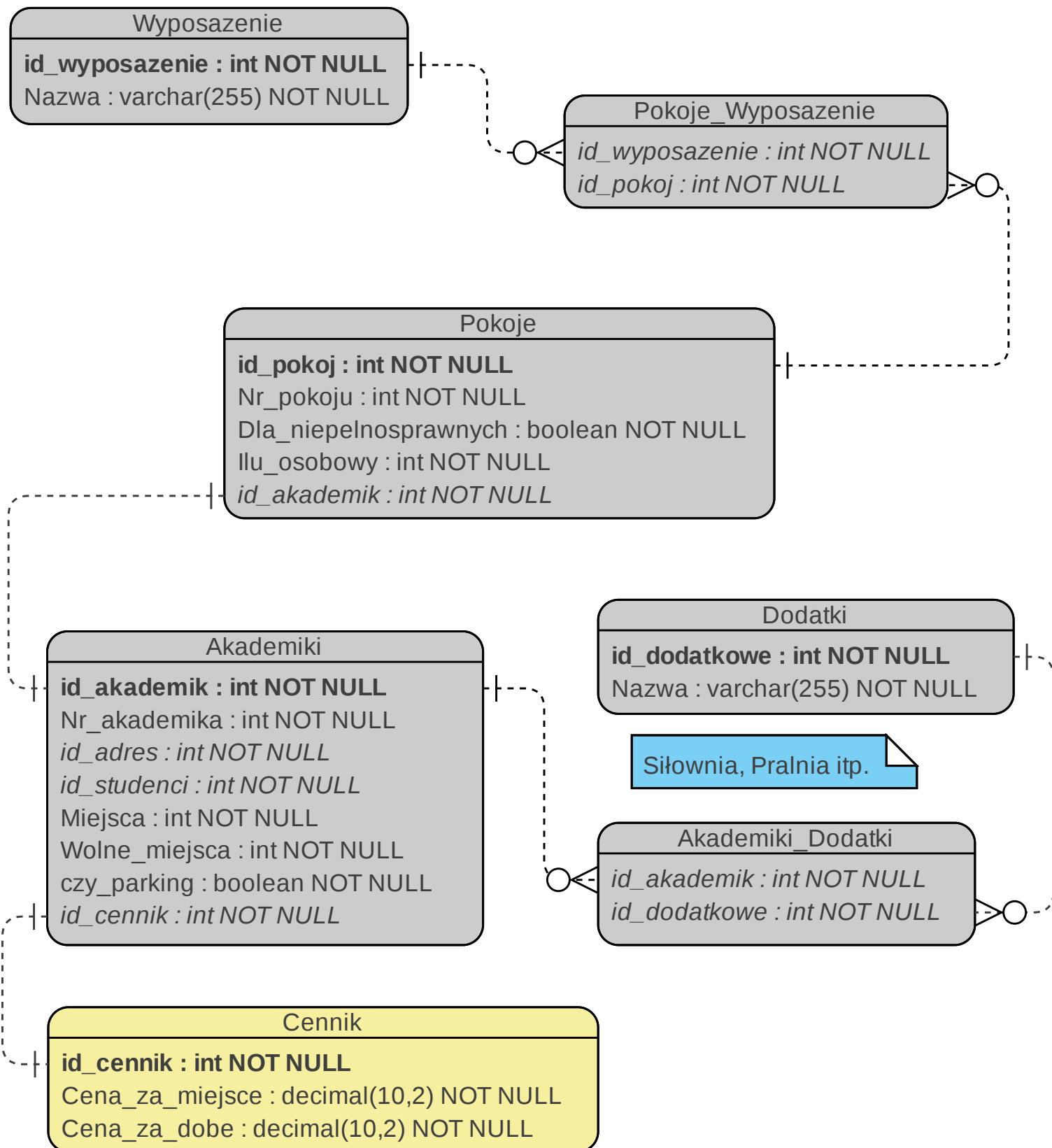
- Mniejszy poziom zaawansowania niż inne systemy takie jak np. PostgreSQL

2.4 Diagram związków encji









3 Przykłady realizacji bazy danych

Poniższe przykłady są w formie tabel SQL uzyskanych za pomocą polecenia DESCRIBE, oraz zapytań SQL i ich wyników.

3.1 Przykłady zawartości najważniejszych tabel

Tabela STUDENCI:

Field	Type	Null	Key	Default	Extra
id_student	int(11)	NO	PRI	NULL	A_I
nr_albumu	int(11)	NO		NULL	
imie	varchar(255)	NO		NULL	
nazwisko	varchar(255)	NO		NULL	
data_urodzenia	date	NO		NULL	
pesel	varchar(11)	NO	UNI	999999999999	
id_adres	int(11)	NO		NULL	
email	varchar(255)	NO		NULL	
telefon	varchar(15)	NO		NULL	
rok_rozpoczecia	int(11)	NO		NULL	

Tabela ADRES:

Field	Type	Null	Key	Default	Extra
id_adres	int(11)	NO	PRI	NULL	A_I
ulica	varchar(255)	NO		NULL	
numer_budynku	varchar(5)	NO		NULL	
numer_mieszkania	varchar(5)	YES		NULL	
kod_pocztowy	varchar(10)	NO		NULL	
miescowosc	varchar(255)	NO		NULL	
kraj	varchar(255)	NO		NULL	

Tabela PRACOWNICY:

Field	Type	Null	Key	Default	Extra
id_pracownik	int(11)	NO	PRI	NULL	A_I
id_adres	int(11)	NO		NULL	
imie	varchar(255)	NO		NULL	
nazwisko	varchar(255)	NO		NULL	
email	varchar(255)	NO		NULL	
telefon	varchar(15)	NO		NULL	
data_zatrudnienia	date	NO		NULL	
id_tytul	int(11)	NO		NULL	
pensja	decimal(10,2)	NO		NULL	

Tabela AKADEMIKI:

Field	Type	Null	Key	Default	Extra
id_akademik	int(11)	NO	PRI	NULL	A_I
nr_akademika	int(11)	NO		NULL	
id_adres	int(11)	NO		NULL	
id_student	int(11)	NO		NULL	
miejsca	int(11)	NO		NULL	
wolne_miejsca	int(11)	NO		NULL	
czy_parking	tinyint(1)	NO		NULL	
id_cennik	int(11)	NO		NULL	

Tabela WYDZIAŁ:

Field	Type	Null	Key	Default	Extra
id_wydzial	int(11)	NO	PRI	NULL	A_I
Nazwa	varchar(255)	NO		NULL	
id_adres	int(11)	NO		NULL	
Telefon	varchar(15)	NO		NULL	
Fax	varchar(15)	NO		NULL	
Strona_www	varchar(255)	NO		NULL	

Tabela OCENY:

Field	Type	Null	Key	Default	Extra
id_ocena	int(11)	NO	PRI	NULL	A_I
id_student	int(11)	NO		NULL	
id_przedmiot	int(11)	NO		NULL	
id_rodzaj_o	int(11)	NO		NULL	
id_ocena_w	int(11)	YES		NULL	
zaliczenie	tinyint(1)	YES		NULL	

3.2 Przykłady kilku zapytań i ich wyników

```
01 | SELECT imie,
02 |        nazwisko,
03 |        nr_albumu
04 | FROM studenci;
```

imie	nazwisko	nr_albumu
Jan	Nowakowski	12301
Adam	Nowak	12302
Anna	Kot	12303
Tomasz	Kowalski	32104
Pawel	Kowal	32105

```
01 | SELECT przedmioty.nazwa AS Przedmiot,
02 |        Round(Avg(wartosci_ocen.nazwa), 2) AS Srednia
03 | FROM oceny
04 |      INNER JOIN studenci
05 |            ON studenci.id_student = oceny.id_student
06 |      INNER JOIN przedmioty
07 |            ON przedmioty.id_przedmiot = oceny.id_przedmiot
08 |      INNER JOIN wartosci_ocen
09 |            ON wartosci_ocen.id_ocena_w = oceny.id_ocena_w
10 | WHERE studenci.nr_albumu = 12303
11 | GROUP BY przedmioty.nazwa;
```

Przedmiot	Srednia
Ewolucja	4.00
Panstwo	4.67

```
01 | SELECT przedmioty.rok_prowadzenia AS Rok,
02 |        przedmioty.nazwa           AS Przedmiot
03 | FROM   przedmioty
04 |        INNER JOIN grupy
05 |            ON grupy.id_grupa = przedmioty.id_grupa
06 |        INNER JOIN studenci
07 |            ON studenci.id_student = grupy.id_student
08 | WHERE  studenci.nr_albumu = 12302
09 | ORDER BY przedmioty.rok_prowadzenia;
```

```
+-----+-----+
| Rok   | Przedmiot          |
+-----+-----+
| 2017  | Historia Nowoczesna |
| 2018  | Historia Starożytna |
| 2019  | Państwo             |
| 2019  | Historia Polski     |
+-----+-----+
```

```
01 | SELECT imie,
02 |        nazwisko,
03 |        pensja
04 | FROM   pracownicy
05 | WHERE  pensja > 5000;
```

```
+-----+-----+-----+
| imie   | nazwisko | pensja |
+-----+-----+-----+
| Kacper | Dabrowski | 6000.00 |
| Patrycja | Wysocka | 8000.00 |
| Ada    | Kowalska | 6500.00 |
| Marcin | Zawadzki | 8200.00 |
+-----+-----+-----+
```