

**Kierunek**: Informatyka

**Specjalność**: Technologie internetowe i bazy danych

**Imię i nazwisko:** Adrian Kut

**Nr** **Albumu**: 23821

**Sale do wynajęcia**

Krosno 28.01.2020

**Spis treści**

[**1.** Założenia projektowe 3](#_Toc31138051)

[**2.** Diagramy ERD 3](#_Toc31138052)

[**3.** Skrypt DDL 6](#_Toc31138055)

[**4.** Przykładowe zapytania na bazie danych 8](#_Toc31138056)

[INSERT INTO: 8](#_Toc31138057)

[SELECT: 9](#_Toc31138058)

[SELECT Z GROUP BY: 9](#_Toc31138059)

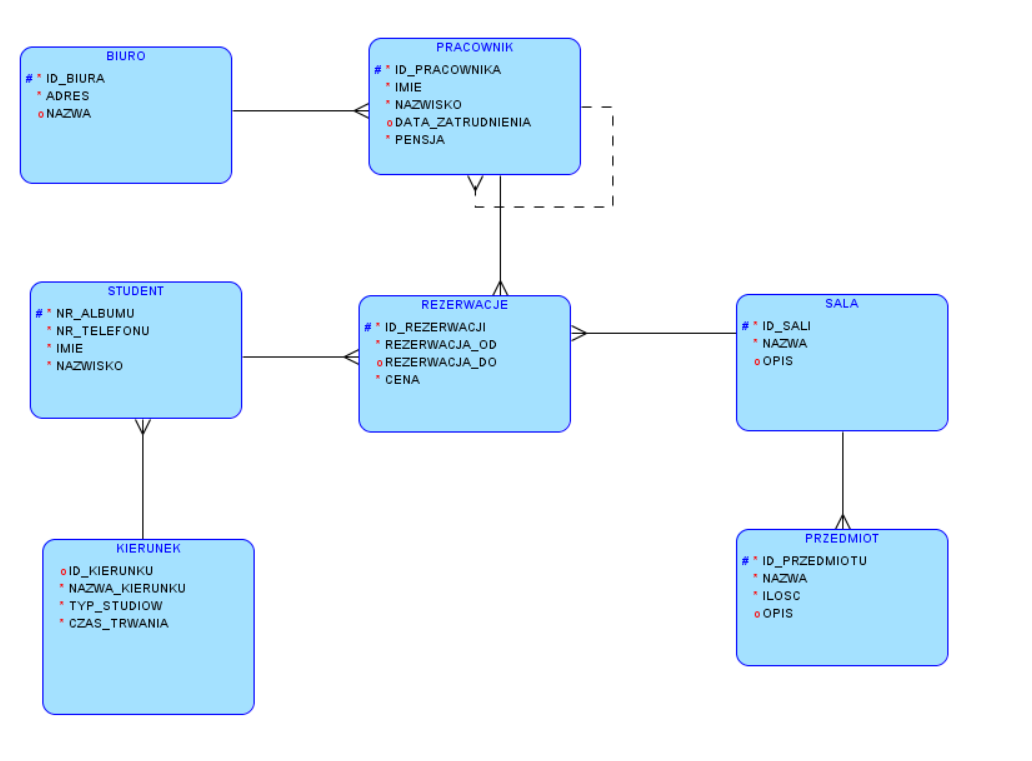
[PODZAPYTANIA: 10](#_Toc31138060)

[PERSPEKTYWY: 10](#_Toc31138061)

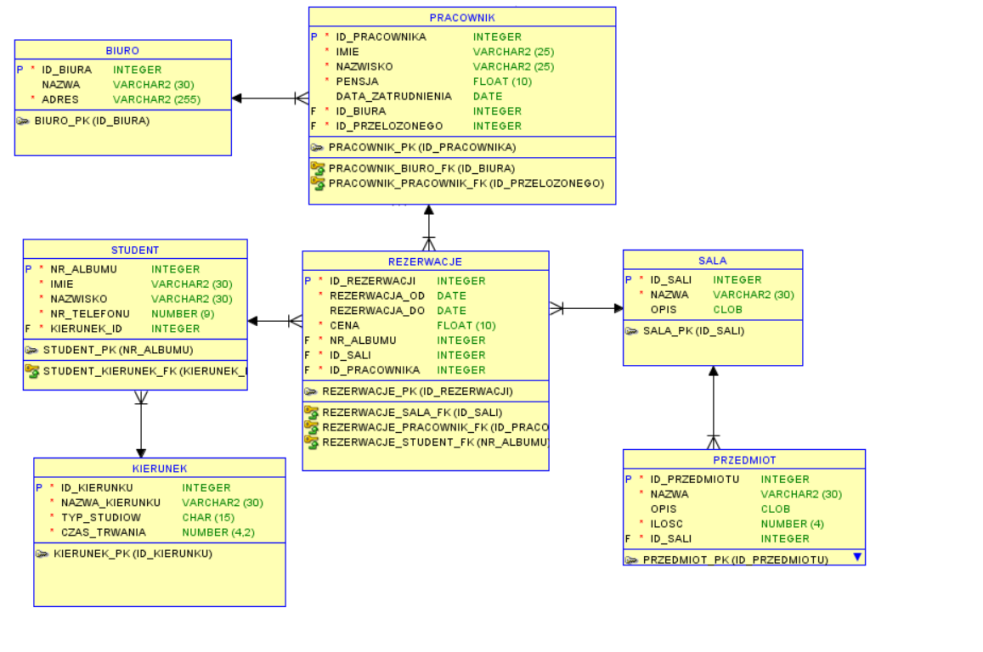
# 1. Założenia projektowe

* Prosta aplikacja do zarządzania wynajmu sali, wyłącznie dla studentów. Niezależnie od kierunku studiów.
* Aplikacja będzie zrealizowana po stronie klienta z interfejsem do wyboru sal z tych, które aktualnie są dostępne, jeśli nie, to otrzyma odpowiednią informację zwrotną, do kiedy będzie zajęta.
* Do dyspozycji będą pomieszczenia przeznaczone do nauki, na małe imprezy okolicznościowe i konferencje.
* Niezależnie od wielkości sali, jedna osoba wynajmuję całą i dopiero w interesie tej osoby jest zebranie należnych pieniędzy za najem. Oczywiście w salach mogą przebywać inne osoby, ale to osoba najmująca bierze odpowiedzialność za jej stan.
* W systemie nie ma informacji o wykładowcach, ponieważ nie są Oni w jakikolwiek sposób wyszczególnieni w systemie.
* Po ukończeniu studiów student jest usuwany z bazy.
* Sprzęt znajduję się już w sali i nie ma potrzeby, aby opisywać żadne magazyny, dostawców.

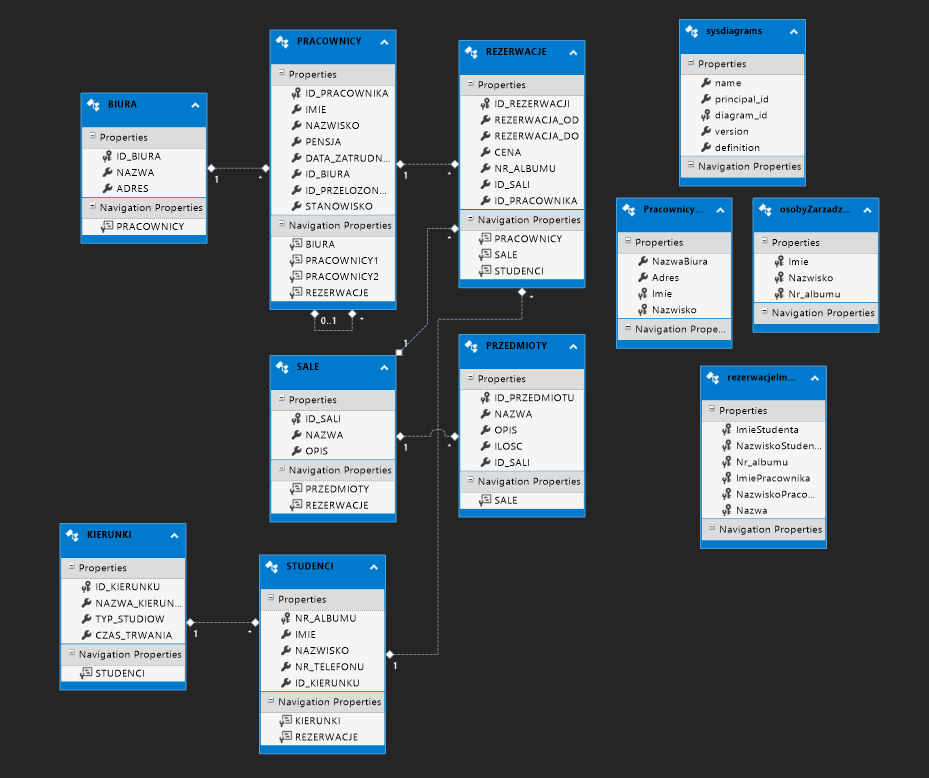
# 2. Diagramy ERD



Rysunek 1 Model logiczny



Rysunek 2 Model Relacyjny



Rysunek 3 Model w aplikacji ASP.NET MVC

# 

Rysunek 4 Główny widok aplikacji

# 

Rysunek 5 Przykładowy widok aplikacji z listą kierunków

# 3. Skrypt DDL

CREATE TABLE BIURA

(

ID\_BIURA INTEGER NOT NULL ,

NAZWA VARCHAR (30) ,

ADRES VARCHAR (255) NOT NULL

) ;

ALTER TABLE BIURA ADD CONSTRAINT BIURO\_PK PRIMARY KEY ( ID\_BIURA ) ;

CREATE TABLE KIERUNKI

(

ID\_KIERUNKU INTEGER NOT NULL ,

NAZWA\_KIERUNKU VARCHAR (30) NOT NULL ,

TYP\_STUDIOW CHAR (15) NOT NULL ,

CZAS\_TRWANIA NUMERIC (4,2) NOT NULL

) ;

ALTER TABLE KIERUNKI ADD CONSTRAINT KIERUNEK\_PK PRIMARY KEY ( ID\_KIERUNKU ) ;

CREATE TABLE PRACOWNICY

(

ID\_PRACOWNIKA INTEGER NOT NULL ,

IMIE VARCHAR (25) NOT NULL ,

NAZWISKO VARCHAR (25) NOT NULL ,

PENSJA FLOAT (10) NOT NULL ,

DATA\_ZATRUDNIENIA DATE DEFAULT SYSDATETIME(),

ID\_BIURA INTEGER NOT NULL ,

ID\_PRZELOZONEGO INTEGER ,

STANOWISKO VARCHAR(25) NULL,

) ;

ALTER TABLE PRACOWNICY ADD CONSTRAINT PRACOWNIK\_PK PRIMARY KEY ( ID\_PRACOWNIKA ) ;

CREATE TABLE PRZEDMIOTY

(

ID\_PRZEDMIOTU INTEGER NOT NULL ,

NAZWA VARCHAr (30) NOT NULL ,

OPIS TEXT ,

ILOSC NUMERIC (4) NOT NULL ,

ID\_SALI INTEGER NOT NULL

) ;

ALTER TABLE PRZEDMIOTY ADD CONSTRAINT PRZEDMIOT\_PK PRIMARY KEY ( ID\_PRZEDMIOTU ) ;

CREATE TABLE REZERWACJE

(

ID\_REZERWACJI INTEGER NOT NULL ,

REZERWACJA\_OD DATE DEFAULT SYSDATETIME() NOT NULL,

REZERWACJA\_DO DATE ,

CENA FLOAT (10) NOT NULL ,

NR\_ALBUMU INTEGER NOT NULL ,

ID\_SALI INTEGER NOT NULL ,

ID\_PRACOWNIKA INTEGER NOT NULL

) ;

ALTER TABLE REZERWACJE ADD CONSTRAINT REZERWACJE\_PK PRIMARY KEY ( ID\_REZERWACJI ) ;

CREATE TABLE SALE

(

ID\_SALI INTEGER NOT NULL ,

NAZWA VARCHAR (30) NOT NULL ,

OPIS TEXT

) ;

ALTER TABLE SALE ADD CONSTRAINT SALA\_PK PRIMARY KEY ( ID\_SALI ) ;

CREATE TABLE STUDENCI

(

NR\_ALBUMU INTEGER NOT NULL ,

IMIE VARCHAR (30) NOT NULL ,

NAZWISKO VARCHAR (30) NOT NULL ,

NR\_TELEFONU numeric (9) NOT NULL ,

ID\_KIERUNKU INTEGER NOT NULL

) ;

ALTER TABLE STUDENCI ADD CONSTRAINT STUDENT\_PK PRIMARY KEY ( NR\_ALBUMU ) ;

ALTER TABLE PRACOWNICY ADD CONSTRAINT PRACOWNIK\_BIURO\_FK FOREIGN KEY ( ID\_BIURA ) REFERENCES BIURA ( ID\_BIURA ) ;

ALTER TABLE PRACOWNICY ADD CONSTRAINT PRACOWNIK\_PRACOWNIK\_FK FOREIGN KEY ( ID\_PRZELOZONEGO ) REFERENCES PRACOWNICY ( ID\_PRACOWNIKA ) ;

ALTER TABLE PRZEDMIOTY ADD CONSTRAINT PRZEDMIOT\_SALA\_FK FOREIGN KEY ( ID\_SALI ) REFERENCES SALE ( ID\_SALI ) ;

ALTER TABLE REZERWACJE ADD CONSTRAINT REZERWACJE\_PRACOWNIK\_FK FOREIGN KEY ( ID\_PRACOWNIKA ) REFERENCES PRACOWNICY ( ID\_PRACOWNIKA ) ;  
ALTER TABLE REZERWACJE ADD CONSTRAINT REZERWACJE\_SALA\_FK FOREIGN KEY ( ID\_SALI ) REFERENCES SALE ( ID\_SALI ) ;  
ALTER TABLE REZERWACJE ADD CONSTRAINT REZERWACJE\_STUDENT\_FK FOREIGN KEY ( NR\_ALBUMU ) REFERENCES STUDENCI ( NR\_ALBUMU ) ;  
ALTER TABLE STUDENCI ADD CONSTRAINT STUDENT\_KIERUNEK\_FK FOREIGN KEY ( ID\_KIERUNKU) REFERENCES KIERUNKI ( ID\_KIERUNKU ) ;

# 4. Przykładowe zapytania na bazie danych

## INSERT INTO:

1. **Kierunki**

Insert into Kierunki (ID\_KIERUNKU,NAZWA\_KIERUNKU,TYP\_STUDIOW,CZAS\_TRWANIA) values (1,'Informatyka','Inzynierskie ',3.5);

1. **Studenci**

Insert into Studenci (NR\_ALBUMU,IMIE,NAZWISKO,NR\_TELEFONU,ID\_KIERUNKU) values (23821,'Adrian','K',667283440,1);

1. **Sale**

Insert into Sale (ID\_SALI,NAZWA,OPIS) values ('1','Konferencyjna','W tej sali mogą odbywać się różnego rodzaju spotkania. Mniej lub bardziej oficjalne.');

1. **Przedmioty**

Insert into PRZEDMIOTY (ID\_PRZEDMIOTU,NAZWA,OPIS,ILOSC,ID\_SALI) values (1,'Kula dyskotekowa','Kula typu disco','1',1);

1. **Biura**

Insert into BIURA (ID\_BIURA,NAZWA,ADRES) values ('1','Glowne','Warszawa ul.Kwiatkowska 32');

Insert into BIURA (ID\_BIURA,NAZWA,ADRES) values ('2','Pracownicze','Krosno ul.Podkarpacka 32')

1. **Pracownicy**  
   Insert into PRACOWNICY (ID\_PRACOWNIKA,IMIE,NAZWISKO,PENSJA,DATA\_ZATRUDNIENIA,ID\_BIURA,ID\_PRZELOZONEGO,STANOWISKO) VALUES (1,'Adam','Szefowski','3500','2003-12-18 14:00:00','1',null,'Kierownik');

Insert into PRACOWNICY (ID\_PRACOWNIKA,IMIE,NAZWISKO,PENSJA,DATA\_ZATRUDNIENIA,ID\_BIURA,ID\_PRZELOZONEGO,STANOWISKO) VALUES (2,'Katarzyna','Cios','2100','2019-12-19 10:00:00','2','1','Recepcjonista');

1. **Rezerwacje**

Insert into REZERWACJE (ID\_REZERWACJI,REZERWACJA\_OD,REZERWACJA\_DO,CENA,NR\_ALBUMU,ID\_SALI,ID\_PRACOWNIKA) VALUES ('1','2019-12-19 11:35:00','2019-12-21 11:35:00','300','23821','2','2');

## SELECT:

1. **Studenci kierunku Informatyka**

select s.Imie, s.Nazwisko, k.Nazwa\_kierunku

from kierunki k

join studenci s on k.id\_kierunku=s.id\_kierunku

where k.nazwa\_kierunku = 'Informatyka';

1. **Przedmioty w sali Konferencyjna**

select s.nazwa as NazwaSali, p.nazwa as NazwaPrzedmiotu

from sale s

join przedmioty p on s.id\_sali = p.id\_sali

where s.nazwa = 'Konferencyjna';

1. **Rezerwacje na sale Imprezową warte wiecej niz 200 zl, z danymi Pracownika i Studenta**

select stu.imie as ImieStudenta,stu.nazwisko as NazwiskoStudenta,r.Cena ,

p.imie as ImiePracownika,p.nazwisko as NazwiskoPracownika,sal.nazwa as NazwaSali

from studenci stu

join rezerwacje r on stu.nr\_albumu = r.NR\_ALBUMU

join pracownicy p on p.id\_pracownika = r.ID\_PRACOWNIKA

join sale sal on sal.id\_sali = r.id\_sali

where cena > 200 and sal.nazwa = 'Imprezowa';

## SELECT Z GROUP BY:

1. **Liczba studentów na danym kierunku**

select Nazwa\_Kierunku, count(\*) as LiczbaStudentów

from studenci

join kierunki on kierunki.ID\_KIERUNKU = STUDENCI.ID\_KIERUNKU

group by NAZWA\_KIERUNKU

order by LiczbaStudentów desc

1. **Liczba przedmiotóww w poszczególnych salach**

select s.NAZWA, count(\*) as LiczbaPrzedmiotow

from przedmioty p

join sale s on s.id\_sali = p.id\_sali

group by s.nazwa

1. **Liczba rezerwacji obsłużonych przez pracownika Katarzyna**

select count(\*) as LiczbaRezerwacji, Imie, Nazwisko

from Pracownicy

where imie = 'Katarzyna'

group by Imie, Nazwisko

## PODZAPYTANIA:

1. **Których przedmiotów jest więcej niż Drukarki 3w1**

select Nazwa,Ilosc from przedmioty

where ILOSC > (select ILOSC from przedmioty where nazwa like 'Drukarka 3w1')

order by ilosc desc;

1. **Pracownicy którzy mają przełożonego**

select \* from pracownicy where id\_pracownika in

(select ID\_PRACOWNIKA from pracownicy where ID\_PRZELOZONEGO is not null)

1. **Liczba kierunków danego typu studiów, które trwają krócej niż kierunek 'Pielegniarstwo'**

select count(\*)as LiczbaKierunków,Typ\_Studiow,Czas\_Trwania from kierunki

where czas\_trwania <

(select czas\_trwania from kierunki where nazwa\_kierunku = 'Pielegniarstwo')  
group by TYP\_STUDIOW,CZAS\_TRWANIA;

## PERSPEKTYWY:

1. **Wszyscy pracownicy z biura 'Pracownicze'**

create view PracownicyPracownicze

as

select b.Nazwa as NazwaBiura, b.Adres, p.Imie, p.Nazwisko

from biura b

right join pracownicy p on p.id\_biura = b.id\_biura

where b.nazwa = 'Pracownicze';

2. **Imiona, nazwiska i numer albumu osób z kierunku Zarządzanie**

create view osobyZarzadzanie

as

select s.Imie, s.Nazwisko, s.Nr\_albumu

from kierunki k

join studenci s on k.id\_kierunku=s.id\_kierunku

where k.nazwa\_kierunku = 'Zarzadzanie';

**3. Rezerwacje na sale Imprezową**

create view rezerwacjeImprezowa as

select stu.Imie as ImieStudenta,stu.Nazwisko as NazwiskoStudenta,stu.Nr\_albumu,

prac.Imie as ImiePracownika, prac.Nazwisko as NazwiskoPracownika, Nazwa

from rezerwacje r

join studenci stu on stu.NR\_ALBUMU = r.NR\_ALBUMU

join pracownicy prac on prac.ID\_PRACOWNIKA = r.ID\_PRACOWNIKA

join sale on sale.ID\_SALI = r.ID\_SALI

where nazwa = 'Imprezowa';