

POLITECHNIKA POZNAŃSKA
WYDZIAŁ ELEKTRYCZNY
Instytut Automatyki, Robotyki i Inżynierii
Informatycznej

Aleksandra Laskowska
Projekt bazy danych LIGA Piłkarska

12 czerwca 2019

LIGA ERD

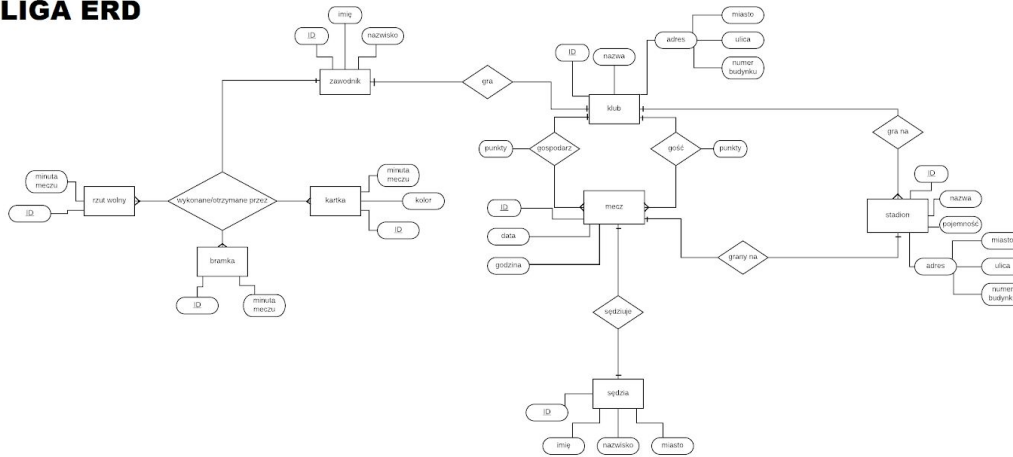


Diagram 1 "Diagram ER bazy danych "Liga""

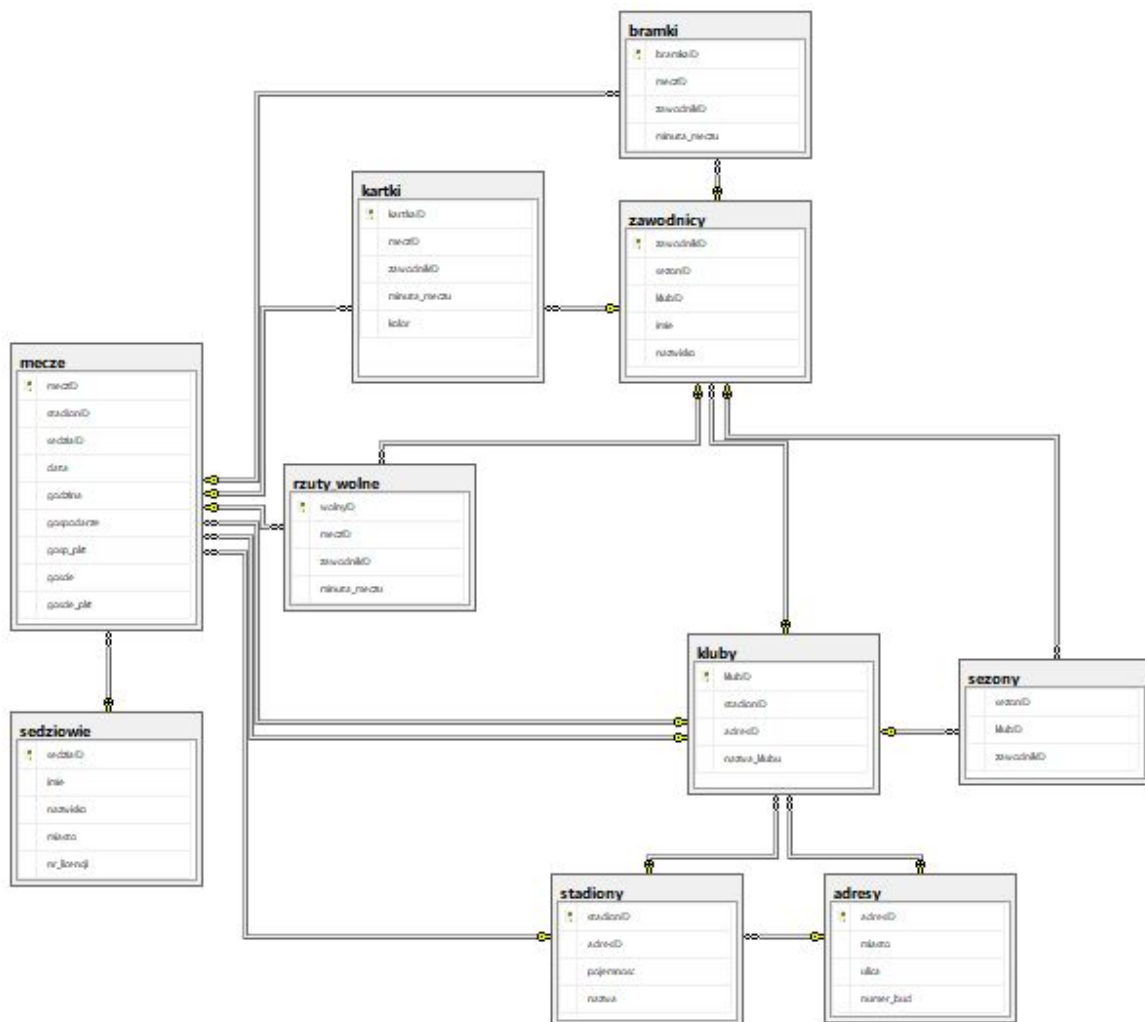


Diagram 2 "Model relacyjny"

SCHEMAT RELACYJNY

Adresy(adresID, miasto, ulica, numer_bud)

Stadiony(stadionID, adresID, pojemnosc, nazwa)
 Kluby(klubID, stadionID, adresID, nazwa_klubu)
 Sezony(sezonID, klubID, zawodnikID)
 Sedziowie(sedzialID, imie, nazwisko, miasto, nr_licencji)
 Mecze(meczID, stadionID, sedzialID, data, godzina, gospodarze, gosp_pkt, goscie, goscie_pkt)
 Zawodnicy(zawodnikID, klubID, imie, nazwisko)
 Kartki(kartkaID, meczID, zawodnikID, minuta_meczu, kolor)
 Rzuty_wolne(wolnyID, meczID, zawodnikID, minuta_meczu)
 Bramki(bramkaID, meczID, zawodnikID, minuta_meczu)

DEFINICJA SCHEMATU RELACYJNEGO W SQL

CREATE TABLE sezony

```
(
    sezonID INT NOT NULL,
    klubID INT NOT NULL FOREIGN KEY REFERENCES kluby(klubID)
    zawodnikID INT NOT NULL FOREIGN KEY REFERENCES
zawodnicy(zawodnikID)
)
```

CREATE TABLE zawodnicy

```
(
    zawodnikID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    sezonID INT NOT NULL FOREIGN KEY REFERENCES sezony(sezonID),
    klubID INT NOT NULL FOREIGN KEY REFERENCES kluby(klubID),
    imie varchar(30) NOT NULL,
    nazwisko varchar(30) NOT NULL
)
```

CREATE TABLE kluby

```
(
    klubID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    stadionID INT NOT NULL FOREIGN KEY REFERENCES
stadiony(stadionID),
    adresID INT NOT NULL FOREIGN KEY REFERENCES adresy(adresID),
    nazwa_klubu varchar(50) NOT NULL
)
```

CREATE TABLE adresy

```
(
    adresID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    miasto varchar(30) NOT NULL,
    ulica varchar(30) NOT NULL,
)
```

```
        numer_bud INT NOT NULL
    )
```

```
CREATE TABLE stadiony
```

```
(
    stadionID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    adresID INT NOT NULL FOREIGN KEY REFERENCES adresy(adresID),
    pojemnosc INT NOT NULL,
    nazwa varchar(50) NOT NULL
)
```

```
CREATE TABLE sedziowie
```

```
(
    sedzialID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    imie varchar(30) NOT NULL,
    nazwisko varchar(30) NOT NULL,
    miasto varchar(30) NOT NULL,
    nr_licencji INT NOT NULL
)
```

```
CREATE TABLE mecze
```

```
(
    meczID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    stadionID INT NOT NULL FOREIGN KEY REFERENCES
stadiony(stadionID),
    sedzialID INT NOT NULL FOREIGN KEY REFERENCES
sedziowie(sedzialID),
    data DATE NOT NULL,
    godzina TIME NOT NULL,
    gospodarze INT NOT NULL FOREIGN KEY REFERENCES kluby(klubID),
    gosp_pkt INT DEFAULT 0,
    goscie INT NOT NULL FOREIGN KEY REFERENCES kluby(klubID),
    goscie_pkt INT DEFAULT 0
)
```

```
CREATE TABLE bramki
```

```
(
    bramkaID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,
    meczID INT NOT NULL FOREIGN KEY REFERENCES mecze(meczID),
    zawodnikID INT NOT NULL FOREIGN KEY REFERENCES
zawodnicy(zawodnikID),
    minuta_meczu TIME NOT NULL
)
```

)

CREATE TABLE rzuty_wolne

(

 wolnyID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

 mecznID INT NOT NULL FOREIGN KEY REFERENCES mecze(mecznID),

 zawodnikID INT NOT NULL FOREIGN KEY REFERENCES

zawodnicy(zawodnikID),

 minuta_meczu TIME NOT NULL

)

CREATE TABLE kartki

(

 kartkaID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

 mecznID INT NOT NULL FOREIGN KEY REFERENCES mecze(mecznID),

 zawodnikID INT NOT NULL FOREIGN KEY REFERENCES

zawodnicy(zawodnikID),

 minuta_meczu TIME NOT NULL,

 kolor varchar(10) NOT NULL

)

TWORZENIE WIDOKU RANKING

CREATE VIEW RANKING

AS

select sum(goscie_pkt) + sum(gosp_pkt) as punkty, nazwa_klubu

from mecze m join kluby k on (m.gospodarze=klubID or m.goscie=k.klubID)

group by nazwa_klubu

TRIGGER NA WIDOK RANKING

ALTER TRIGGER NowyMecz

on RANKING

INSTEAD OF INSERT

AS

BEGIN

 DECLARE @pkt_gosp INT

 DECLARE @pkt_gosc INT

 SELECT @pkt_gosp=gosp_pkt

from mecze

join inserted on

inserted.nazwa_klubu=mecze.gospodarze

```

SELECT @pkt_gosc=goscie_pkt
from mecze
join inserted on
inserted.nazwa_klubu=mecze.goscie

if(not(
(@pkt_gosc=3 and @pkt_gosp=0) or
(@pkt_gosc=1 and @pkt_gosp=1) or
(@pkt_gosc=0 and @pkt_gosp=3) ))
BEGIN
RAISERROR('Punkty w meczu sa blednie przyznane',16,1)
return
END

INSERT INTO mecze (gosp_pkt,goscie_pkt)
SELECT @pkt_gosp,@pkt_gosc
from inserted

END

```

SPRAWDZENIE, CZY PODANY CZAS MECZU JEST POPRAWNY

```

alter table kartki
ADD constraint czasKartka
check (minuta_meczu >= '00:00' and minuta_meczu <= '01:30')

alter table bramki
ADD constraint czasBramka
check (minuta_meczu >= '00:00' and minuta_meczu <= '01:30')

alter table rzuty_wolne
ADD constraint czasWolny
check (minuta_meczu >= '00:00' and minuta_meczu <= '01:30')

```

WYBRANE ZAPYTANIA

-- liczba bramek strzelonych przez zawodników każdego klubu

```
select count(*) as liczbabramek, nazwa_klubu from bramki join mecze on  
bramki.meczID=mecze.meczID  
join kluby on (mecze.gospodarze=kluby.klubID or mecze.goscie=kluby.klubID)  
group by nazwa_klubu
```

--suma punktów zdobyta w sezonie przez każdy klub

```
select sum(goscie_pkt) + sum(gosp_pkt) as punkty, nazwa_klubu  
from mecze m join kluby k on (m.gospodarze=klubID or m.goscie=k.klubID)  
group by nazwa_klubu  
order by punkty desc
```

--nazwiska, imiona i nazwy klubów zawodników, którzy zdobyli czerwone kartki

```
select nazwisko, imie, nazwa_klubu from zawodnicy, kartki, kluby  
where zawodnicy.zawodnikID=kartki.zawodnikID and kartki.kolor='czerwona' and  
zawodnicy.klubID=kluby.klubID
```

--nazwiska, imiona i nazwy klubów zawodników, którzy zdobyli żółte kartki

```
select nazwisko, imie, nazwa_klubu from zawodnicy, kartki, kluby  
where zawodnicy.zawodnikID=kartki.zawodnikID and kartki.kolor='żółta' and  
zawodnicy.klubID=kluby.klubID
```

--lista drużyn w kolejności od największej ilości rzutów wolnych

```
select count(*) as liczbarzutow, nazwa_klubu from rzuty_wolne, kluby, zawodnicy  
where rzuty_wolne.zawodnikID=zawodnicy.zawodnikID and  
zawodnicy.klubID=kluby.klubID  
group by nazwa_klubu order by liczbarzutow desc
```

--wyświetlenie wszystkich rzutów wolnych dla danego klubu

```
select wolnyID, nazwa_klubu, nazwisko from rzuty_wolne, zawodnicy, kluby  
where rzuty_wolne.zawodnikID=zawodnicy.zawodnikID and  
zawodnicy.klubID=kluby.klubID  
order by nazwa_klubu
```