POLITECHNIKA POZNAŃSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

Karol Sienkiewicz
nr indeksu 140774

Projekt bazy danych

Grupa L7

02 Lipca 2020

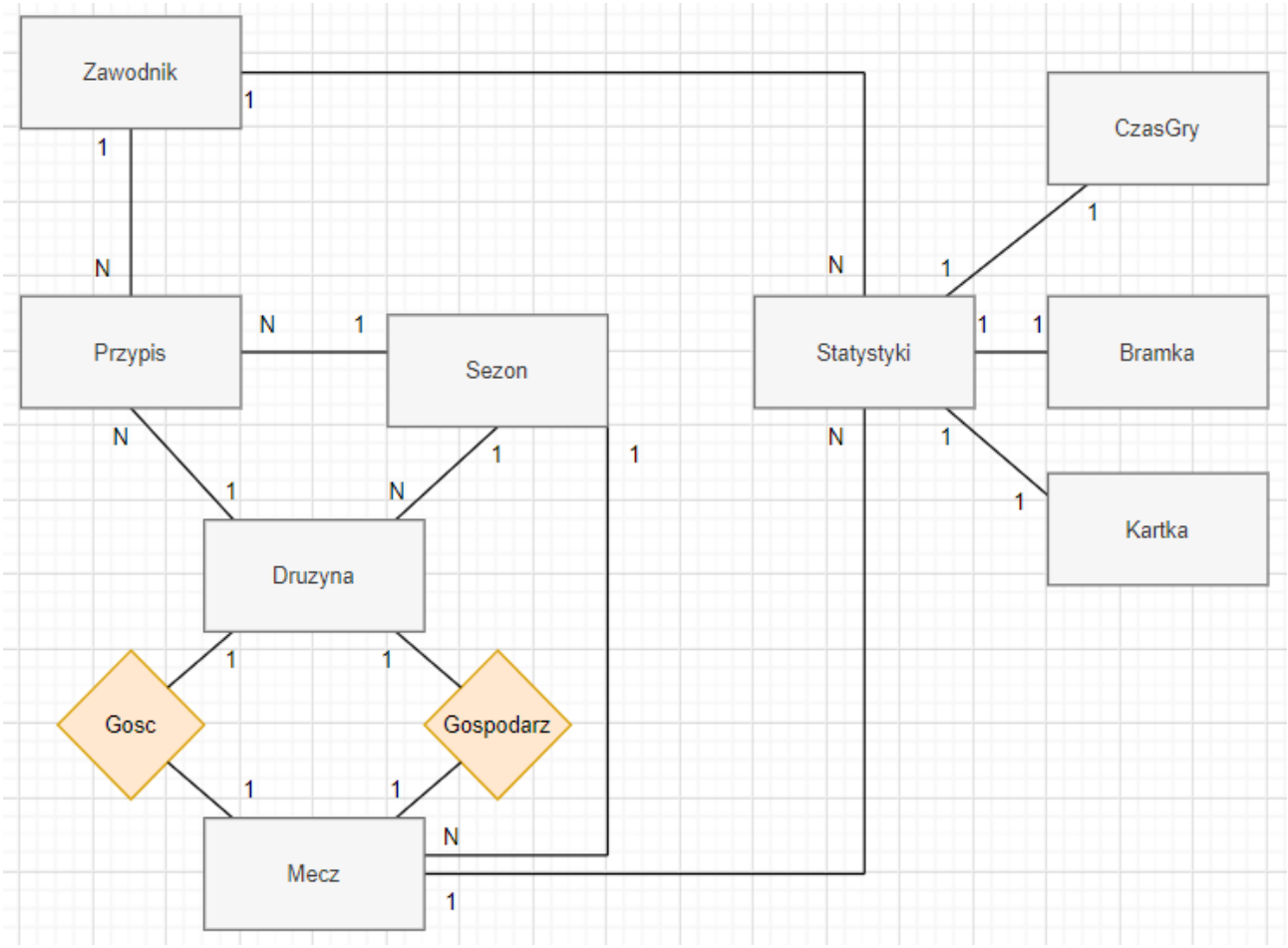
Wymagania

- zawodnicy przypisani są do drużyn: w jednym sezonie przypisanie jest stałe – w różnych sezonach może być różne,
- drużyny w sezonie rozgrywają dwa mecze: w każdym meczu biorą udział dwie drużyny, jedna pełni rolę gościa, a druga gospodarza,
- rejestrowane są dane o przebiegu każdego meczu: a) kto grał i w jakim czasie, b) kto i kiedy strzelił bramkę, c) kto i kiedy otrzymał żółtą/czerwoną kartkę

Dla bazy danych:

- opracuj diagram (obiektowo-związkowy) ER,
- podaj wynik transformacji do schematu relacyjnego,
- wymień warunki spójności, które dotyczą bazy danych LIGA; podaj sposób definiowania poszczególnych warunków spójności: w definicji tabeli, jako reguła czy jako trigger;
- napisz procedurę wyzwalaną, która po zakończeniu meczu wyznacza aktualną tabelę rankingową drużyn oraz listę strzelców bramek.

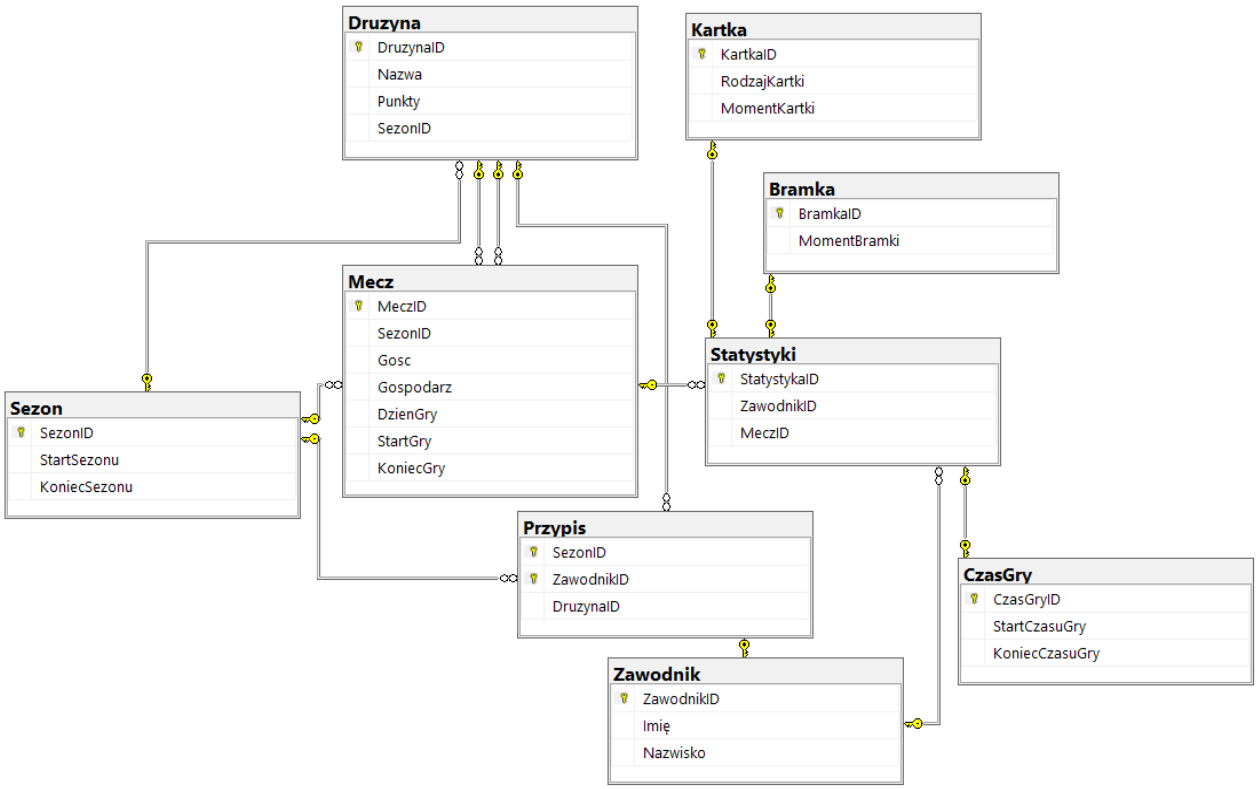
Diagram ER



Wyjaśnienie encji:

Zawodnik to encja opisująca danego piłkarza.
Przypis zawiera informacje o tym, który piłkarz grał w jakiej drużynie podczas danego sezonu.
Sezon zawiera informacje o sezonie.
Druzyna opisuje drużynę grającą w danym sezonie.
Mecz zawiera informacje o rozegranym meczu.
Statystyki przechowują listę wszystkich statystyk.
CzasGry opisuje czas gry zawodnika.
Bramka zawiera informacje o strzelonej bramce.
Kartka zawiera informacje o tym jakiego rodzaju kartkę dostał zawodnik.

Transformacja do modelu relacyjnego



Uwagi do modelu:
Relacje Gosc oraz Gospodarz zostały zaimplementowane jako atrybuty w tabeli Mecz.

Warunki integralności

W tabeli Przypis klucz główny jest złożony z id zawodnika (ZawodnikID z tabeli Zawodnik) oraz id sezonu (SezonID z tabeli Sezon) co zapewnia, że dany zawodnik nie może w jednym sezonie występować w wielu drużynach.
Wykorzystano głównie integralność encji (czyli stosowanie klucza głównego, który gwarantuje, że jedna krotka nie wystąpi wielokrotnie w tabeli) oraz integralności odwołań (poprzez stosowanie kluczy obcych jednoznacznie odwołujących się do krotek w innych tabelach).

Warunki zostały zdefiniowane podczas tworzenia tabeli:

Zawodnik

```
CREATE TABLE Zawodnik (
ZawodnikID int NOT NULL IDENTITY(0,1) PRIMARY KEY,
Imię varchar(60) NOT NULL,
Nazwisko varchar(60) NOT NULL
)
```

Sezon

```
CREATE TABLE Sezon (
SezonID int NOT NULL IDENTITY(1,1) PRIMARY KEY,
StartSezonu date NOT NULL,
KoniecSezonu date
)
```

Druzyna

```
CREATE TABLE Druzyna (
DruzynaID int NOT NULL IDENTITY(0,1) PRIMARY KEY,
Nazwa varchar(60) NOT NULL,
Punkty int DEFAULT 0,
SezonID int NOT NULL
FOREIGN KEY (SezonID) REFERENCES Sezon(SezonID)
)
```

Przypis

```
CREATE TABLE Przypis (
SezonID int NOT NULL
FOREIGN KEY (SezonID) REFERENCES Sezon(SezonID),
ZawodnikID int NOT NULL
FOREIGN KEY (ZawodnikID) REFERENCES Zawodnik(ZawodnikID),
DruzynaID int NOT NULL
FOREIGN KEY (DruzynaID) REFERENCES Druzyna(DruzynaID),
CONSTRAINT PrzypisID PRIMARY KEY(ZawodnikID,SezonID)
)
```

Mecz

```
CREATE TABLE Mecz (
MeczID int NOT NULL IDENTITY(0,1) PRIMARY KEY,
SezonID int NOT NULL
FOREIGN KEY (SezonID) REFERENCES Sezon(SezonID),
Gosc int NOT NULL
FOREIGN KEY (Gosc) REFERENCES Druzyna(DruzynaID),
Gospodarz int NOT NULL
FOREIGN KEY (Gospodarz) REFERENCES Druzyna(DruzynaID),
DzienGry date NOT NULL,
StartGry time NOT NULL,
KoniecGry time
)
```

Statystyki

```
CREATE TABLE Statystyki (
StatystykaID int NOT NULL IDENTITY(0,1) PRIMARY KEY,
ZawodnikID int NOT NULL
FOREIGN KEY (ZawodnikID) REFERENCES Zawodnik(ZawodnikID),
MeczID int NOT NULL
FOREIGN KEY (MeczID) REFERENCES Mecz(MeczID)
)
```

CzasGry

```
CREATE TABLE CzasGry (
CzasGryID int NOT NULL
FOREIGN KEY (CzasGryID) REFERENCES Statystyki(StatystykaID),
StartCzasuGry time NOT NULL,
KoniecCzasuGry time NOT NULL,
PRIMARY KEY (CzasGryID)
)
```

Bramka

```
CREATE TABLE Bramka (
BramkaID int NOT NULL
FOREIGN KEY (BramkaID) REFERENCES Statystyki(StatystykaID),
PRIMARY KEY (BramkaID),
MomentBramki time NOT NULL,
)
```

Kartka

```
CREATE TABLE Kartka (
KartkaID int NOT NULL
FOREIGN KEY (KartkaID) REFERENCES Statystyki(StatystykaID),
RodzajKartki varchar(8) NOT NULL
CHECK (RodzajKartki LIKE 'czerwona' OR RodzajKartki LIKE 'zolta'),
PRIMARY KEY (KartkaID),
MomentKartki time NOT NULL,
)
```

Trigger

W celu uproszczenia triggera zaimplementowano dwie poniższe metody:

Strzelcy

```
CREATE OR ALTER PROCEDURE Strzelcy @IDSezonu int
AS
SELECT za.Imię, za.Nazwisko, GoleSezonu.IloscGoli
FROM Zawodnik za
JOIN
(SELECT z.ZawodnikID, COUNT(*) AS IloscGoli
FROM Zawodnik z
JOIN
(SELECT s.ZawodnikID
FROM Statystyki s
JOIN Bramka b
ON b.BramkaID = s.StatystykaID
WHERE s.MeczID IN
(SELECT m.MeczID
FROM Mecz m
WHERE m.SezonID = @IDSezonu)) AS StatystykiSezonu
ON z.ZawodnikID = StatystykiSezonu.ZawodnikID
GROUP BY z.ZawodnikID) AS GoleSezonu
ON za.ZawodnikID = GoleSezonu.ZawodnikID
ORDER BY GoleSezonu.IloscGoli DESC
go
```

RankingDruzyn

```
CREATE OR ALTER PROCEDURE RankingDruzyn @IDSezonu int
AS
SELECT d.Nazwa, d.Punkty
FROM Druzyna d
WHERE d.SezonID = @IDSezonu
ORDER BY d.Punkty DESC
go
```

Natomiast trigger wyzwalany na koniec meczu wygląda następująco:

KoniecMeczu

```
CREATE OR ALTER TRIGGER KoniecMeczu
ON Mecz
FOR UPDATE AS
	DECLARE @MeczID int
	DECLARE @SezonID int
	DECLARE @Gosc int
	DECLARE @Gospodarz int
	DECLARE @KoniecGry time
	DECLARE @GoleGosc int
	DECLARE @GoleGospodarz int
	SET @MeczID = (SELECT MeczID FROM inserted)
	SET @SezonID = (SELECT SezonID FROM inserted)
```

```
	SET @Gosc = (SELECT Gosc FROM inserted)
	SET @Gospodarz = (SELECT Gospodarz FROM inserted)
	SET @KoniecGry = (SELECT KoniecGry FROM inserted)
	SET @GoleGosc =
		(SELECT COUNT(*) AS GoleDruzyny
		FROM Mecz m
		JOIN Statystyki s
		ON m.MeczID = s.MeczID
		JOIN Bramka b
		ON s.StatystykaID = b.BramkaID
		JOIN Zawodnik z
		ON z.ZawodnikID = s.ZawodnikID
		JOIN Przypis p
		ON p.ZawodnikID = z.ZawodnikID
		WHERE m.MeczID = @MeczID AND p.SezonID = m.SezonID AND m.Gosc = p.DruzynaID
		GROUP BY p.DruzynaID)
	SET @GoleGospodarz =
		(SELECT COUNT(*) AS GoleDruzyny
		FROM Mecz m
		JOIN Statystyki s
		ON m.MeczID = s.MeczID
		JOIN Bramka b
		ON s.StatystykaID = b.BramkaID
		JOIN Zawodnik z
		ON z.ZawodnikID = s.ZawodnikID
		JOIN Przypis p
		ON p.ZawodnikID = z.ZawodnikID
		WHERE m.MeczID = @MeczID AND p.SezonID = m.SezonID AND m.Gospodarz =
p.DruzynaID
		GROUP BY p.DruzynaID)
	IF @GoleGosc IS NULL
		BEGIN
		SET @GoleGosc = 0
		END

	IF @GoleGospodarz IS NULL
		BEGIN
		SET @GoleGospodarz = 0
		END

	IF @KoniecGry IS NOT NULL
		BEGIN
		IF(@GoleGosc = @GoleGospodarz)
			BEGIN
			UPDATE Druzyna
			SET Punkty = Punkty +1
			WHERE DruzynaID = @Gosc OR DruzynaID = @Gospodarz
			END
		ELSE IF(@GoleGosc > @GoleGospodarz)
			BEGIN
			UPDATE Druzyna
			SET Punkty = Punkty +3
			WHERE DruzynaID = @Gosc
			END
		ELSE IF(@GoleGosc < @GoleGospodarz)
			BEGIN
			UPDATE Druzyna
			SET Punkty = Punkty +3
			WHERE DruzynaID = @Gospodarz
			END
		EXEC RankingDruzyn @IDSezonu = @SezonID
		EXEC Strzelcy @IDSezonu = @SezonID
		END
```