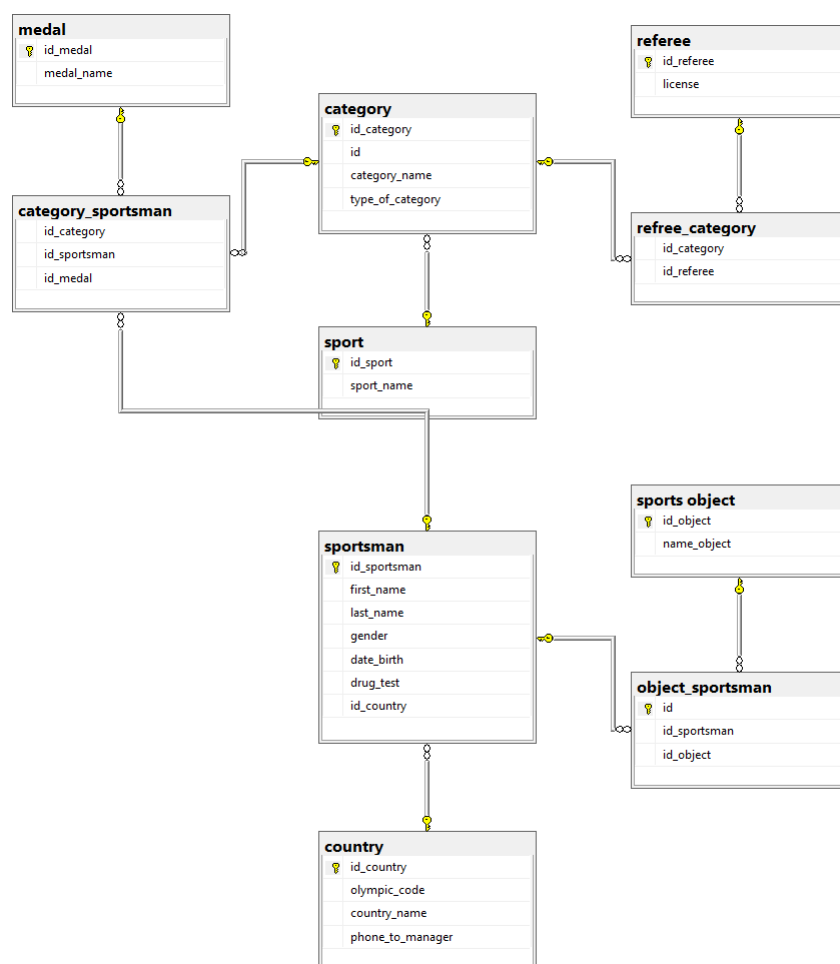


# Olympics - projekt na przedmiot bazy danych

Justyna Gruba

February 2021

## 1 Diagram ER



## 2 Tworzenie tabel

```
1 --dyscyplina sportowa
2 CREATE TABLE sport (
3 id_sport INT NOT NULL PRIMARY KEY,
4 sport_name VARCHAR(255) NULL);
5
6 --kategoria dyscypliny sportowej
7 CREATE TABLE category (
8 id_category INT NOT NULL PRIMARY KEY,
9 id INT NULL REFERENCES sport(id_sport),
10 category_name VARCHAR(40) NULL,
11 type_of_category VARCHAR(10) NULL);
12
13 --obiekty sportowe takie jak stadiony czy korty tenisowe
14 CREATE TABLE [sports object] (
15 id_object INT NOT NULL PRIMARY KEY,
16 name_object VARCHAR(200));
17
18 --kraje, z kt rych pochodz sportowcy
19 CREATE TABLE country (
20 id_country INT NOT NULL PRIMARY KEY,
21 olympic_code VARCHAR(14),
22 country_name VARCHAR(20),
23 phone_to_manager CHAR(12),
24 CONSTRAINT phone CHECK (phone_to_manager NOT LIKE '+%[^0-9]%'));
25
26 --tabela sportowc w
27 CREATE TABLE sportsman (
28 id_sportsman INT NOT NULL PRIMARY KEY,
29 first_name VARCHAR(45) NOT NULL,
30 last_name VARCHAR(45) NOT NULL,
31 gender VARCHAR(1),
32 date_birth DATETIME NOT NULL,
33 drug_test VARCHAR(6) NULL,
34 id_country INT REFERENCES country(id_country));
35
36 -- ka dy sportowiec mo e wyst powa na kilku obiektach
37 CREATE TABLE object_sportsman (
38 id INT PRIMARY KEY,
39 id_sportsman INT REFERENCES sportsman(id_sportsman),
40 id_object INT REFERENCES [sports object](id_object));
41
42 -- wygrane medale podczas zawod w
43 CREATE TABLE medal(
44 id_medal INT PRIMARY KEY,
45 medal_name VARCHAR(10));
46
47 --ka dy sportowiec mo e bra udzia w wi cej ni jednej
   kategorii
48 CREATE TABLE category_sportsman(
49 id_category INT REFERENCES category(id_category),
50 id_sportsman INT REFERENCES sportsman(id_sportsman),
51 id_medal INT REFERENCES medal(id_medal));
52
53 --tabela s dzi w
54 CREATE TABLE referee(
```

```

55 id_referee INT PRIMARY KEY,
56 license VARCHAR(3));
57
58 --ka dy s dzia mo e s dziowa wi cej ni jedn kategori
59 CREATE TABLE referee_category(
60 id_category INT REFERENCES category(id_category),
61 id_referee INT REFERENCES referee(id_referee));
62 GO

```

### 3 Tworzenie procedur

```

1 --procedura sprawdza czy wpisany kraj ma komitet olimpijski
2 CREATE PROCEDURE noc @country VARCHAR(15)
3 AS
4 IF NOT EXISTS(SELECT country_name FROM country WHERE country_name =
   @country)
5 BEGIN
6     RAISERROR('Nie ma takiego kraju w bazie danych',16,1)
7     RETURN(1)
8 END
9 IF EXISTS(SELECT olympic_code FROM country WHERE @country =
   country_name AND olympic_code IS NULL)
10 BEGIN
11     RAISERROR('Podany kraj nie ma komitetu olimpijskiego',16,1)
12     RETURN(2)
13 END
14
15 ELSE
16     SELECT *
17     FROM country
18     WHERE country_name = @country
19     RETURN(0)
20 GO

```

```

1 --procedura, ktora ustawia typ dla danej kategorii np. basketball -
   'team'
2 CREATE PROCEDURE type_category (@id_category INT)
3 AS
4 DECLARE @type VARCHAR(7)
5 SET @type = (SELECT type_of_category FROM category WHERE
   @id_category = id_category)
6 DECLARE @id INT
7 SET @id = (SELECT id FROM category WHERE id_category = @id_category
   )
8 IF @id = 9
9 BEGIN
10     UPDATE category
11     SET type_of_category = 'team'
12     WHERE @id = id
13 END
14 ELSE
15 BEGIN
16     UPDATE category
17     SET type_of_category = 'single'
18     WHERE @id = id

```

```

19     END
20 GO

```

## 4 Tworzenie wyzwalaczy

```

1  --wyzwalacz, ktory sprawdza czy dodani sportowcy zdali test
   antydopingowy
2  CREATE TRIGGER test_drug ON sportsman
3  AFTER INSERT
4  AS
5  DECLARE @test VARCHAR(6)
6  SET @test = (SELECT drug_test FROM inserted)
7  IF @test LIKE 'failed'
8  BEGIN
9      ROLLBACK
10     RAISERROR('Uczestnik usuniety z powodu dopingu',16,1)
11 END
12 IF @test LIKE 'passed'
13 BEGIN
14     PRINT('Test antydopingowy zdany')
15 END
16 IF @test LIKE NULL
17 BEGIN
18     ROLLBACK
19     RAISERROR('Powtorzyc test i dodac uczestnika jeszcze raz'
20     ,16,1)
21 END
22 GO

```

```

1  --wyzwalacz, ktory sprawdza czy sportowiec ma odpowiedni wiek
2  CREATE TRIGGER birth_check ON sportsman
3  AFTER INSERT
4  AS
5  DECLARE @year INT
6  SET @year = (SELECT YEAR(date_birth) FROM inserted)
7  DECLARE @current_year INT
8  SET @current_year = YEAR(GETDATE())
9  DECLARE @age INT
10 SET @age = @current_year - @year
11 IF @age >= 16
12 BEGIN
13     PRINT('Poprawny wiek na uczestnictwo w Olimpiadzie')
14 END
15 ELSE
16 BEGIN
17     ROLLBACK
18     RAISERROR('Nie mo na doda uczestnika do bazy danych',16,1)
19 END
20 GO

```

## 5 Tworzenie funkcji

```

1  --funkcja zlicza medale sportowcow, kt rzy je zdobyli
2  CREATE FUNCTION count_medals(@medal VARCHAR(8))
3  RETURNS TABLE
4  AS
5  RETURN(
6  SELECT s.id_sportsman, s.full_name, COUNT(*) AS medal
7  FROM sportsman AS s
8  INNER JOIN object_sportsman AS os
9  ON os.id_sportsman = s.id_sportsman
10 INNER JOIN category_sportsman AS cs
11 ON cs.id_sportsman = os.id_sportsman
12 INNER JOIN medal AS m
13 ON cs.id_medal = m.id_medal
14 WHERE medal_name= @medal
15 GROUP BY s.id_sportsman, s.full_name);
16 GO
17
18 SELECT * FROM count_medals('Gold')

1  --wyswietla wszystkie kategorie dla danego sportu
2  CREATE FUNCTION categories(@nazwa_sportu VARCHAR(15))
3  RETURNS TABLE
4  AS
5  RETURN(
6  SELECT id_category, category_name
7  FROM category AS c
8  JOIN sport AS s
9  ON s.id_sport = c.id
10 WHERE @nazwa_sportu = s.sport_name);
11 GO
12
13 SELECT * FROM categories('Volleyball')

1  --wyswietla sportowcow, ktorzy odbywaj swoje zawody na danym
   obiekcie sportowym
2  CREATE FUNCTION objects_sportsman(@objects VARCHAR(15))
3  RETURNS TABLE AS
4  RETURN(
5  SELECT s.full_name, c.category_name
6  FROM object_sportsman AS os
7  JOIN sportsman AS s
8  ON s.id_sportsman = os.id_sportsman
9  JOIN [sports object] AS so
10 ON os.id_object = so.id_object
11 JOIN category_sportsman AS cs
12 ON cs.id_sportsman = os.id_sportsman
13 JOIN category AS c
14 ON c.id_category = cs.id_category
15 WHERE @objects = so.name_object)
16 GO
17 SELECT * FROM objects_sportsman('Stadion')

```

## 6 Tworzenie widoków

```

1  --widok pokazuje kategorie w kt rych s dziowie przyznali medale
2  CREATE VIEW medals_referee AS
3  SELECT r.id_referee, m.medal_name, c.category_name
4  FROM referee AS r
5  JOIN referee_category AS rc
6
7  ON r.id_referee = rc.id_referee
8  JOIN category_sportsman AS sc
9
10 ON rc.id_category = sc.id_category
11 JOIN medal AS m
12 ON sc.id_medal = m.id_medal
13 JOIN category AS c
14 ON c.id_category = sc.id_category
15 WHERE r.license LIKE 'TAK'
16 AND m.id_medal = 1 OR m.id_medal = 2 OR m.id_medal = 3
17 GO
18
19 SELECT * FROM license

```

```

1  --pokazuje wszystkich sedziow z aktualna licencja
2  CREATE VIEW license AS
3  SELECT r.id_referee, c.category_name
4  FROM referee AS r
5  JOIN referee_category AS rc
6  ON r.id_referee = rc.id_referee
7  JOIN category_sportsman AS sc
8  ON rc.id_category = sc.id_category
9  JOIN category AS c
10 ON c.id_category = sc.id_category
11 WHERE r.license LIKE 'TAK'
12 GO
13
14 SELECT * FROM license

```

```

1  --wypisuje dla kazdego kraju ile jest sportowcow danej plci
2  CREATE VIEW country_gender AS
3  SELECT c.country_name, s.gender, COUNT(s.gender) AS NUMBER
4  FROM sportsman AS s, country AS c, sportsman_country AS sc
5  WHERE s.id_sportsman = sc.id_sportsman AND c.id_country = sc.
   id_country
6  GROUP BY c.country_name, s.gender
7  GO

```