

SQL protokol – evidence Ultimate Frisbee

Michaela Mašková

9. listopadu 2020

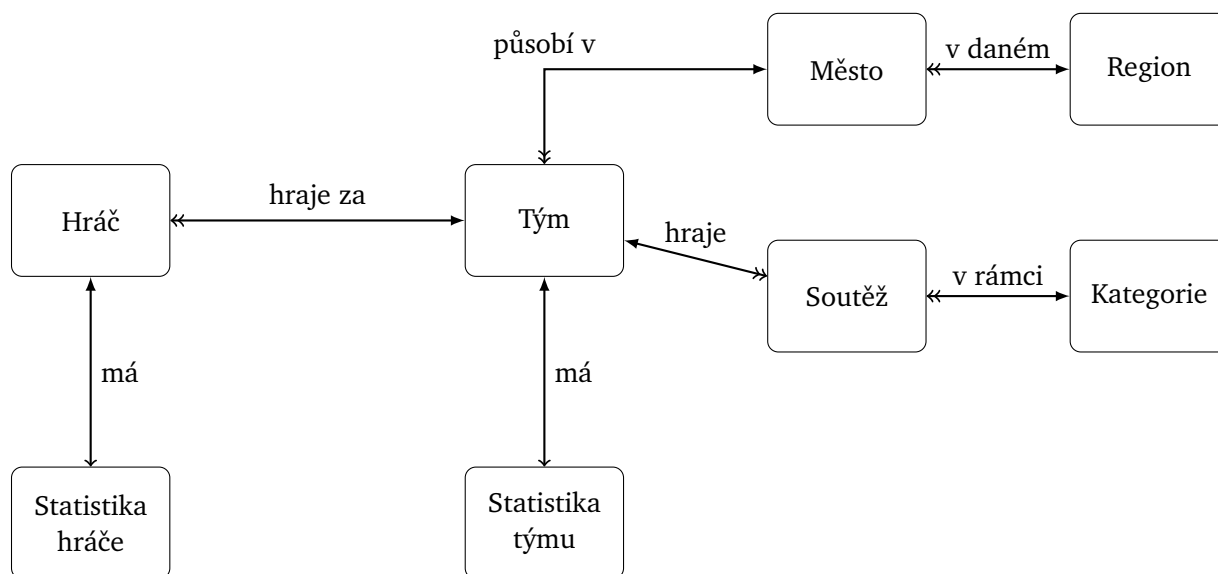
Představení problému

Tato databáze má za cíl mapovat českou ultimate frisbee scénu jako evidenci jejích členů společně s nezbytnými informacemi o týmech. Zahrnuje taktéž statistiky jak hráče, tak týmu.

Pár poznámek:

- Česká scéna je rozdělená na několik regionů (spádových oblastí). Týmy tak působí v jednotlivých regionech a zároveň mají své domovské město.
- Ultimate frisbee se hraje ve třech kategoriích: mix, muži a ženy.
- Spirit of the game je zvláštní hodnocení fair play, které má celkem 5 kategorií, v nichž se udává hodnocení mezi 0 a 4. Součet těchto hodnot je celkový výsledek spirita. Spirit se udává vždy pro každý tým za každý odehraný turnaj.
- Produktivita hráče se počítá jako počet bodů (skóre + asistence) v průměru na zápas.

Schéma



Tabulky

Přehled

Hráč	Tým	Město	Region
<ul style="list-style-type: none"> • rodné číslo • jméno • příjmení • číslo dresu • tym.id 	<ul style="list-style-type: none"> • id.tym • název týmu • mesto.id 	<ul style="list-style-type: none"> • id.mesto • název města • region.id 	<ul style="list-style-type: none"> • id.region • název regionu

Soutěž	Kategorie	Statistika týmu	Statistika hráče
<ul style="list-style-type: none"> • tym.id • kategorie.id 	<ul style="list-style-type: none"> • id.kategorie • název kategorie 	<ul style="list-style-type: none"> • tym.id • znalost pravidel • fauly a body contact • férové smýšlení • pozitivní přístup • komunikace 	<ul style="list-style-type: none"> • hrac.id • skóre • asistence • počet odehraných zápasů

Vytvoření tabulek

```

1
2  CREATE TABLE tym(
3      tymID INTEGER PRIMARY KEY,
4      nazev_tymu TEXT,
5      mestoID INTEGER,
6      FOREIGN KEY (mestoID) REFERENCES mesto(mestoID)
7  );
8
9  CREATE TABLE hrac(
10     rc INTEGER PRIMARY KEY,
11     jmeno TEXT,
12     prijmeni TEXT,
13     cislo INTEGER,
14     tymID INTEGER,
15     FOREIGN KEY (tymID) REFERENCES tym(tymID)
16 );
17
18 CREATE TABLE mesto(
19     mestoID INTEGER PRIMARY KEY,
20     nazev_mesta TEXT,
21     regionID INTEGER,
22     FOREIGN KEY (regionID) REFERENCES region(regionID)
23 );
24
25
26
27

```

```
28 CREATE TABLE region(  
29     regionID int PRIMARY KEY,  
30     nazev_regionu text  
31 );  
32  
33 CREATE TABLE kategorie(  
34     kategorieID INTEGER PRIMARY KEY,  
35     nazev_kategorie TEXT  
36 );  
37  
38 CREATE TABLE soutez(  
39     tymID INTEGER,  
40     kategorieID INTEGER,  
41     FOREIGN KEY (tymID) REFERENCES tym(tymID),  
42     FOREIGN KEY (kategorieID) REFERENCES kategorie(kategorieID)  
43 );  
44  
45 CREATE TABLE hostovani(  
46     hracID INTEGER,  
47     tymID INTEGER,  
48     FOREIGN KEY (hracID) REFERENCES hrac(hracID),  
49     FOREIGN KEY (tymID) REFERENCES tym(tymID)  
50 );  
51  
52 CREATE TABLE statistika_tymu(  
53     tymID INTEGER,  
54     pravidla REAL,  
55     fauly REAL,  
56     ferocest REAL,  
57     pozitiv REAL,  
58     komunikace REAL,  
59     FOREIGN KEY (tymID) REFERENCES tym(tymID)  
60 );  
61  
62 CREATE TABLE statistika_hrace(  
63     hracID INTEGER,  
64     skore REAL,  
65     asistence REAL,  
66     zapasu REAL,  
67     FOREIGN KEY (hracID) REFERENCES hrac(rc)  
68 );
```

Views

5 nejproduktivnějších hráčů

Cílem je zobrazit 5 hráčů s nejvyšším skóre produktivity a zároveň ukázat, z jakého pochází týmu. Stejně tak chceme zobrazit i danou statistiku – tedy počet bodů, asistencí i odehraných zápasů.

```

1  CREATE VIEW nejproduktivnejsiHraci AS
2  SELECT
3      jmeno AS Jméno,
4      prijmeni AS Příjmení,
5      (skore + asistence) / zapasu AS Produktivita,
6      nazev_tymu AS Tým,
7      skore AS Skóre,
8      asistence AS Asistence,
9      zapasu AS 'Odehraných zápasů'
10 FROM hrac
11 INNER JOIN statistika_hrace ON statistika_hrace.hracID = hrac.rc
12 INNER JOIN tym ON hrac.tymID = tym.tymID
13 ORDER BY Produktivita DESC
14 LIMIT 5;
```

Kolik hráčů hraje v jednotlivých týmech a v jednotlivých městech?

Cílem je získat views, které vytvoří tabulku týmů společně s aktivním počtem hráčů. Stejná tabulka pak pro jednotlivá města, kde ovšem chceme jak počet hráčů hrajících v jednotlivých městech, tak počet týmů v daném městě.

```

1  CREATE VIEW hracuVTymu AS
2  SELECT
3      nazev_tymu AS Tým,
4      count(rc) AS 'Počet hráčů'
5  FROM hrac
6  INNER JOIN tym ON hrac.tymID = tym.tymID
7  GROUP BY nazev_tymu
8  ORDER BY count(rc) DESC;
```

```

1  CREATE VIEW hracuVeMeste AS
2  SELECT
3      nazev_mesta AS Město,
4      count(rc) AS 'Počet hráčů',
5      count(DISTINCT tym.tymID) AS 'Počet týmů'
6  FROM tym
7  INNER JOIN hrac ON hrac.tymID = tym.tymID
8  INNER JOIN mesto ON mesto.mestoID = tym.mestoID
9  GROUP BY nazev_mesta
10 ORDER BY 'Počet hráčů' DESC;
```

Spirit skóre pro týmy

První view slouží k ukázání všech týmů a jejich průměrného Spiritu. Jelikož vyšší Spirit znamená lepší výsledek, chceme je seřadit sestupně.

```

1  CREATE VIEW tymySpiritAll AS
2  SELECT
3      nazev_tymu AS Tým,
4      pravidla+fauly+ferovost+pozitiv+komunikace AS Spirit,
5      nazev_mesta AS Město,
6      nazev_regionu AS Region
7  FROM tym
8  INNER JOIN statistika_tymu ON statistika_tymu.tymID = tym.tymID
9  INNER JOIN mesto ON mesto.mestoID = tym.mestoID
10 INNER JOIN region ON mesto.regionID = region.regionID
11 ORDER BY Spirit DESC;
```

Následující view vyprodukuje tabulku týmů jednotlivých regionů s nejvyšším Spirit skóre.

```

1  CREATE VIEW tymyNejvyssiSpiritRegionu AS
2  SELECT
3      nazev_tymu AS Tým,
4      pravidla+fauly+ferovost+pozitiv+komunikace AS Spirit,
5      nazev_mesta AS Město,
6      nazev_regionu AS Region
7  FROM tym
8  INNER JOIN statistika_tymu ON statistika_tymu.tymID = tym.tymID
9  INNER JOIN mesto ON mesto.mestoID = tym.mestoID
10 INNER JOIN region ON mesto.regionID = region.regionID
11 GROUP BY nazev_regionu
12 HAVING max(Spirit);
```

Které týmy hrají ve všech třech kategoriích?

Každý tým může hrát v alespoň jedné nebo až třech kategoriích. Tento view slouží k ukázání týmů, které hrají ve všech třech kategoriích.

```

1  CREATE VIEW tymyVsechKategorii AS
2  SELECT
3      nazev_tymu AS Tým
4  FROM tym
5  INNER JOIN soutez ON soutez.tymID = tym.tymID
6  INNER JOIN kategorie ON kategorie.kategorieID = soutez.kategorieID
7  GROUP BY nazev_tymu
8  HAVING count(nazev_tymu) = 3;
```

Procedury

Následují procedury pro zjednodušení různých úkonů v databázi. Rozdělené jsou na procedury pro hráče a pro týmy.

Procedury pro hráče

První procedura vrátí produktivitu pro hráče daného jména. Druhá pak domovský tým tohoto hráče.

```

1      CREATE PROCEDURE produktivitaHrace @jmeno TEXT, @prijmeni TEXT
2      AS
3      BEGIN
4          SELECT
5              jmeno AS jmeno,
6              prijmeni AS prijmeni,
7              (skore + asistence) / zapasu AS produktivita,
8          FROM hrac
9          INNER JOIN statistika_hrace ON statistika_hrace.hracID = hrac.rc
10         INNER JOIN tym ON hrac.tymID = tym.tymID
11         WHERE jmeno = @jmeno AND prijmeni = @prijmeni;
12     END
13
14     EXEC produktivitaHrace @jmeno = Michal, @prijmeni = David

```

```

1      CREATE PROCEDURE tymHrace @jmeno TEXT, @prijmeni TEXT
2      AS
3      BEGIN
4          SELECT jmeno, prijmeni, tym
5          FROM hrac
6          INNER JOIN tym ON hrac.tymID = tym.tymID
7          WHERE jmeno = @jmeno AND prijmeni = @prijmeni;
8      END
9
10     EXEC tymHrace @jmeno = Michal, @prijmeni = David

```

Poslední procedura přidá nového hráče do tabulky.

```

1      CREATE PROCEDURE pridejHrace @jmeno TEXT, @prijmeni TEXT, @rc INTEGER, @tym TEXT
2      AS
3      BEGIN
4          SELECT tymID FROM tym WHERE nazev_tymu = @tym INTO @newID
5          INSERT INTO hrac (jmeno, prijmeni, rc, tymID)
6              VALUES (@jmeno, @prijmeni, @rc, @newID)
7          COMMIT;
8      END
9
10     EXEC pridejHrace 'Petr', 'Partl', 9606138714, 'Prague_Devils'

```

Procedury pro týmy

Následuje procedura, která pro vybraný tým spočítá jeho průměrný spirit.

```

1  CREATE PROCEDURE spiritTymu @nazev_tymu TEXT
2  AS
3  BEGIN
4      SELECT
5          nazev_tymu AS jmeno,
6          pravidla+fauly+ferovost+pozitiv+komunikace AS spirit,
7      FROM tym
8      INNER JOIN statistika_tymu ON statistika_tymu.tymID = tym.tymID
9      WHERE jmeno = @nazev_tymu;
10 END
11
12 EXEC spiritTymu 'Chupacabras'
```

Poslední procedura je přidání daného týmu.

```

1  CREATE PROCEDURE pridejTym @nazev_tymu TEXT, @nazev_mesta TEXT
2  AS
3  BEGIN
4      IF EXISTS(SELECT * FROM tym) THEN
5          SELECT MAX(tymID)+1 FROM tym INTO @newTymID;
6      ELSE
7          @newTymID=1;
8      END
9      SELECT mestoID FROM mesto WHERE nazev_mesta = @nazev_mesta INTO @newMestoID
10 INSERT INTO tym (tymID, nazev_tymu, mestoID)
11     VALUES (@newTymID, @nazev_tymu, @newMestoID)
12 COMMIT;
13 END
14
15 EXEC pridejTym 'Flying_Tigers', 'Brno'
```