

Csapatsportok projekt

Benedek László (*IH1RZJ, h253741*)

Specifikáció

A feladat egy olyan alkalmazás elkészítése, amely csapatsportok esetén képes a csapatokhoz és azok tagjaihoz kapcsolódó adatok tárolására. Továbbá a csapatok között játszott mérkőzések is a rendszer részét képezik. Az alkalmazás használata legyen regisztrációhoz kötött oly módon, hogy az egyes funkciók csak a regisztrált felhasználó bejelentkezése után legyenek elérhetők.

Tervezés

A feladat által eltárolni kívánt adatok

- Felhasználó: név, felhasználónév, jelszó
- Csapat: név, székhely, alapítás dátuma
- Játékos: név, állampolgárság, születési dátum, poszt
- Mérkőzés: eredmény, dátum, helyszín

Plusz eltárolt adatok

- Pontszerzés: játékos, csapat, mérkőzés
- Hely: ország, város, utca, szám

Feltételezések az adatokról

- Egy játékos egyszerre csak egy csapatban játszhat, de nem biztos, hogy jelenleg van csapata.
- Egy mérkőzésen pontosan két csapat vesz részt és egy csapat több mérkőzésen is részt vehet.
- Egy helynek legalább az országot meg kell határoznia.
- Több azonos nevű és alapítási dátumú csapat is lehet egy városban.
- Több azonos nevű, születési dátumú és nemzetiségű játékos is lehet.
- Egy csapatnak egyszerre csak egy székhelye lehet.
- Egy mérkőzés csak egy helyen lehet egyszerre.
- Ha egy meccsen mind a két csapat ugyan annyi pontot szerez, akkor az győzelemnek számít mind a két csapat számára.

Egyedek

- Felhasználó
- Csapat
- Játékos
- Mérkőzés
- Hely

Kapcsolatok

- Helye: csapat székhelye vagy mérkőzés helye
- Tagja: játékos egy csapat tagja
- Kiáll: két csapat egymás ellen
- Pontszerzés: egy csapatnak egy játékos egy meccsen pontot szerez

Kulcsok

Indokolt mesterséges azonosítók bevezetése, mivel több gyenge egyedünk is van és a lekérdezésekben való egyszerű hivatkozást is egyszerűsítik majd.

EK diagram

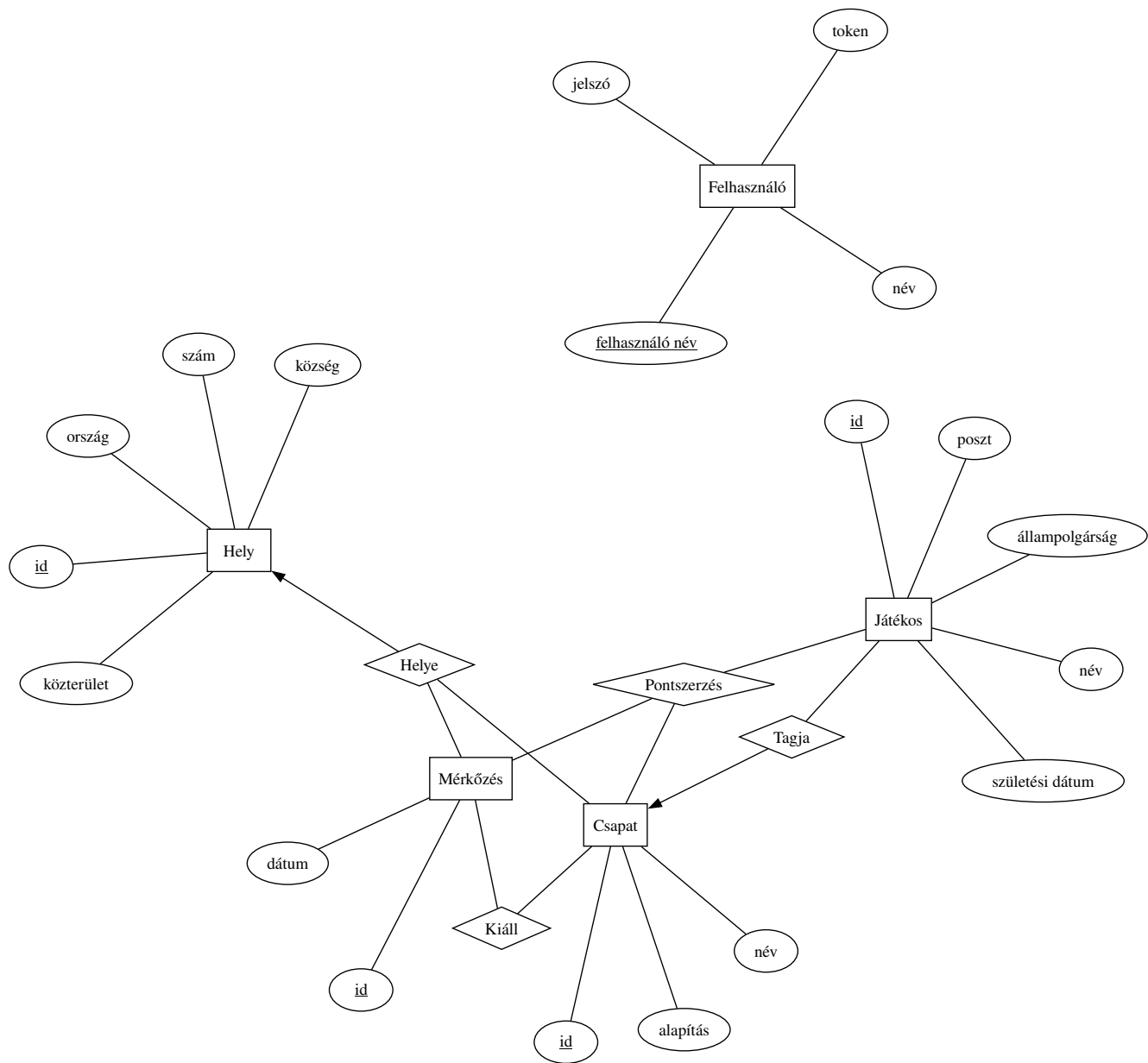


Figure 1: EK diagram

Sémák

location

Megnevezés	Típus	Megjegyzés
id	int	azonosító, kulcs, auto increment
country	varchar(255)	ország
city	varchar(255)	város, lehet NULL
street	varchar(255)	utca, lehet NULL
number	varchar(255)	hászám, lehet NULL

team

Megnevezés	Típus	Megjegyzés
id	int	azonosító, kulcs, auto increment
name	varchar(255)	csapat neve
base	int	székhely, külső kulcs (<i>location</i>)
foundation	date	alapítás dátuma

player

Megnevezés	Típus	Megjegyzés
id	int	azonosító, kulcs, auto increment
name	varchar(255)	név
nationality	varchar(255)	származása
birth_date	date	születési dátum
position	varchar(255)	poszt
team	int	csapat, külső kulcs (<i>team</i>)

match

Megnevezés	Típus	Megjegyzés
id	int	azonosító, kulcs, auto increment
date	date	dátum
location	int	helye, külső kulcs (<i>location</i>)
teamA	int	1. csapat, külső kulcs (<i>team</i>)
teamB	int	2. csapat, külső kulcs (<i>team</i>)

score

Megnevezés	Típus	Megjegyzés
id	int	azonosító, kulcs, auto increment
player	int	játékos, külső kulcs (<i>player</i>)
team	int	csapat, külső kulcs (<i>team</i>)
match	int	mérkőzés, külső kulcs (<i>match</i>)

user

Megnevezés	Típus	Megjegyzés
name	varchar(255)	teljes néve
username	varchar(255)	felhasználónév, kulcs
password	varchar(255)	jelszó bcrypt hash
token	varchar(255)	session token, NULL, ha nincs bejelentkezve

Normálformák

- 1NF: [OK] egyik attribútum se többértékű
- 2NF: [OK] minden egyed kulcsa egy attribútumból áll
- 3NF: [OK] minden attribútum csak a kulcstól függ

Funkciók megvalósítása

Minden egyedhez tartozik egy `src/api/endpoints/[EGYED]/*.go` és egy `src/api/database/[EGYED]/*.go`, ami az ahhoz tartozó funkciók (felvétel, módosítás, törlés) backend részét valósítja meg.

A speciális funkciók a `src/api/endpoints/special/*.go` és a `src/api/database/special/*.go` fájlokban vannak megvalósítva.

A szükséges SQL lekérdezések megtalálhatóak az egyedhez tartozó `src/api/sql/[EGYED]/*.sql` fájlokban.

A megvalósítások frontend része a `src/static/[EGYED]/[FUNKCIÓ]/index.html` és `src/static/[EGYED]/[FUNKCIÓ]/*.js` fájlokban találhatóak.

A backenden minden funkcióhoz egy JSON API tartozik. A módosítást végző funkciók POST, a csak lekérdezést végzők GET metódussal érhetőek el a `src/api/endpoints/endpoints.go` fájlban leírtak szerint.

A szükséges paramétereket a funkcióhoz tartozó `src/api/endpoints/*.go` fájl tetején található struktúra írja le. Az API válaszát a funkcióhoz tartozó `src/api/database/*.go` fájl tetején lévő struktúra határozza meg.

Összetett lekérdezések

- legalább két táblát kell összekapcsolni és van benne csoportosítás és összesítés
 - mérkőzések kilistázása: `src/api/sql/match/get_matches_with_summary.sql`
 - játékosok kilistázása: `src/api/sql/player/get_players_with_summary.sql`
 - csapatok statisztikája: `src/api/sql/special/team_stats.sql`
- allekérdezést tartalmaz
 - mérkőzések kilistázása: `src/api/sql/match/get_matches_with_summary.sql`
 - játékosok kilistázása: `src/api/sql/player/get_players_with_summary.sql`
 - csapatok statisztikája: `src/api/sql/special/team_stats.sql`
 - legrégebben alapított csapat játékosai: `src/api/sql/special/oldest_team.sql`

Telepítés

Docker környezetben

A projekthez tartozik egy Docker Compose projekt is, amiben a szükséges szolgáltatások vannak leírva.

A `api` szolgáltatás saját container image-t használ, ezt futtatás előtt elő kell állítani: `docker compose build api`. A projekt elindítható a `docker/run.sh` futtatásával. Indításnál lehetséges, hogy az API szerver hamarabb elindul, mint a MySQL, ezért az többszöri is újraindulhat, mire a szolgáltatások elérhetőek lesznek.

Szolgáltatások

- mysql: adatbázis szerver
- adminer: adatbázis admin felülete
- nginx: webszerver
- api: backend API szerver

Lokális környezetben

A projekt lefordításához szükségesek a Go nyelvhez tartozó fordítói környezet eszközei. A fordítás a `go build` paranccsal lehetséges a `src/api` könyvtárból.

A `src/static` könyvtár tartalmát egy webszerveren ki kell szolgálni és a `src/static/js/common.js` fájlban az API szerver címét megfelelően kell beállítani.

A program felparaméterezhető környezeti változók, parancssori paraméterek és konfigurációs fájl által is.

változók

Név	Leírás	Alapértelmezett érték
<code>mysql_user</code>	MySQL szerverhez kapcsolódó felhasználó	<code>root</code>
<code>mysql_passwd</code>	MySQL szerverhez kapcsolódó felhasználó jelszava	<code>root</code>
<code>mysql_addr</code>	MySQL szerver címe	<code>mysql:3306</code>
<code>mysql_net</code>	MySQL szerver kapcsolat típusa	<code>tcp</code>
<code>mysql_dbname</code>	MySQL adatbázis neve	<code>app</code>
<code>mysql_allow_native_passwords</code>	MySQL szerver kapcsolat bejelentkezési módja	<code>true</code>
<code>api_addr</code>	API szerver címe	<code>:8080</code>

Használat

Az applikáció felülete böngésző használatával a `http://localhost/` címen érhető el.

Itt először a bejelentkezést nem igénylő funkciókhoz tudunk eljutni. Ha a *Login* oldalon bejelentkezőnk, akkor a többi funkció is elérhetővé válik a *Home* oldalon.

Regisztráció és bejelentkezés

A *Register* oldalon a név, felhasználónév és a jelszó kétszeri megadásával lehet regisztrálni. A rendszer jelzi, hogy a regisztráció sikeres volt-e vagy sem.

A bejelentkezés a *Login* oldalon lehetséges a regisztrációnál megadott felhasználónév és jelszó megadásával. Ha sikeres a bejelentkezés, akkor átirányításra kerülünk a *Home* oldalra.

Kilistázás

Minden kilistázási funkciónál szerepelnek mezők, amik a kilistázandó adatok szűrésére szolgálnak.

A **Filter** gombra kattintva a jelennek meg a szűrés eredményei.

Van olyan eset, hogy egy másik egyed alapján szeretnénk szűrni, például a *játékosokat* a csapatjuk alapján. Ilyenkor először a csapatot kell kiválasztani szűrővel és a **Select** felírra kattintással. Utána lejjebb megadhatjuk a játékos saját adatainak szűrését is.

Hozzáadás

Új adat felvételekor megadhatjuk a kapcsolódó egyedeket a kilistázáshoz hasonló többlépcsős szűrővel, majd lejjebb megadhatjuk a hozzáadni kívánt egyed saját adatit is. Az **Add** gombra való kattintás esetén megtörténik a felvétel és ha nem volt hiba, akkor a kilistázás oldalára irányít át.

Módosítás

Ha be vagyunk jelentkezve, akkor kilistázás esetén a sorokban megjelenik egy **Edit** felirat. Ha erre kattintunk, akkor a kiválasztott bejegyzés adatait módosíthatjuk az új adat felvételéhez hasonló módon. Fontos, hogy azokat az adatokat, amiket nem kívánunk módosítani hagyjuk az eredeti állapotukban.

Törlés

Ugyan azon az oldalon érhetjük el, mint a módosítás funkciót. Ha törölni szeretnénk, akkor a **Remove** gombra kattintva megtehetjük. Ilyenkor még egy utolsó lehetőségünk van arra, hogy meggondoljuk magunkat.

Speciális funkciók

- *Elkövetkező hét meccsei*
- *5 legfiatalabb játékos*
- *Győzelmek és vereségek száma csapatonként*
- *Legrégbben alapított csapat játékosai*
- *Játékosok száma nemzetiség szerint*

Hibák jelzése

Ha valamelyik funkció hibába ütközik futás közben, akkor az *Error* oldalra leszünk átirányítva, ahol arról értesülünk, hogy milyen típusú hiba történt.

Adatbázis közvetlen hozzáférése

Ha szeretnénk ellenőrizni, hogy a kért műveletek ténylegesen megtörténtek-e, akkor azt megtehetjük a <http://localhost/admin/> oldalon. Itt egy Adminer nevű adatbáziskezelő fut. Az alapértelmezett bejelentkezési adatok: *root:root*