Név: Vrabély Gellért
Neptun kód: RWQQ36

Adatbázisok gyakorlat – Tervezési dokumentáció

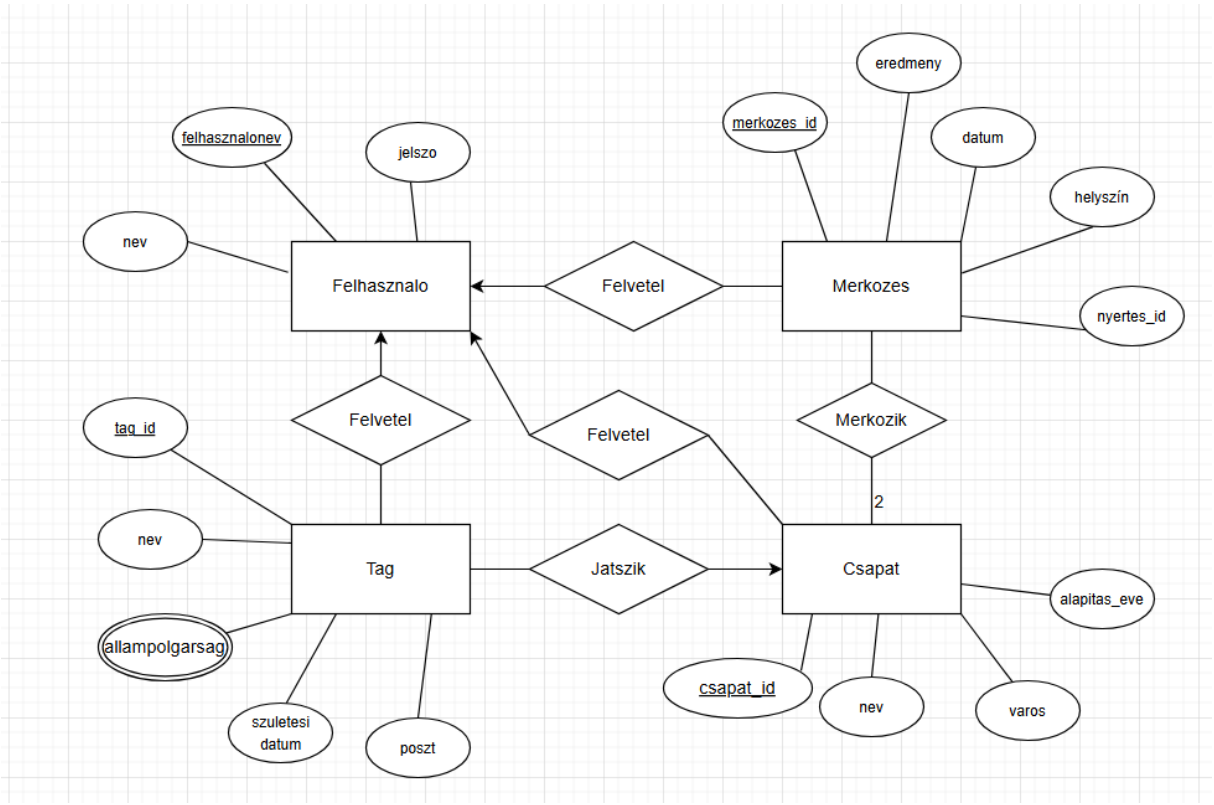
Feladat: Csapatsport

Feladat leírása: 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.

Tárolt adatok (nem feltétlen jelentenek önálló táblákat):
- Felhasználó: felhasználónév, jelszó, név
- Csapat: név, város, alapítás éve
- Tagok: név, állampolgárság, születési dátum, poszt
- Mérkőzések: eredmény, dátum, helyszín

Relációk az adatok között: Egy tag egyszere csak egy csapatban játszhat. Egy mérkőzésen két csapat vesz részt (egy-egy csapat játszik egymás ellen). Egy csapat több mérkőzésen is részt vehet és egy mérkőzésen két csapat vesz részt.

Egyed-kapcsolat modell



Leírás:

- Minden entitást Felhasznalo entitás tud felvenni az adatbázisba, egy felhasználó több entitást is felvehet, de az entitást csak egy felhasználó viheti fel (1 – N kapcsolat), és egy felhasználót a felhasználóneve egyértelműen azonosít

- A Tag entitást a „tag_id” attribútum határozza meg, kapcsolatban áll még a Csapat entitással. Egy Tag egy csapatban játszik, de egy csapatban több tag is játszik. (1 – N kapcsolat)
- A Csapat entitást az egyszerűség kedvéért a csapat_id attribútum határozza meg, kapcsolatban áll az eddigi felsoroltak mellett a Merkozes entitással, egy csapat több mérkőzést is játszhat, és egy mérkőzést 2 csapat játszik, a hazai_e meghatározza, hogy egy csapat hazaiként vagy vendégként játsza a mérkőzést, ez alapján tudjuk meg, kié volt a több pont, így a kapcsolat (2 – N) 2 a többhöz típusú.
- A Merkozes entitást meghatározhatnánk külső kulccsal is, de a Merkozik kapcsolat miatt egyszerűbb egy hozzáadott, merkozes_id attribútumot kulcsként használni.
- A Merkozes entitás „nyertes_id” attribútuma azt mondja el, hogy melyik csapat nyert

Relációs adatbázisséma

Felhasznalo (felhasznalonev, nev, jelszo)
Tag (tag_id, nev, szuletesi datum, poszt, *felhasznalonev, csapat_id*)
Tag_allampolgarsag (tag_id, allampolgarsag, *felhasznalonev, csapat_id*)
Csapat (csapat_id, nev, varos, alapitas_eve, *felhasznalonev*)
Merkozes (merkozes_id, eredmeny, datum, helyszin, *nyertes_id*(csapat_id-ra), *felhasznalonev*)
Merkozik (*csapat_id*, *merkozes_id*)

Normalizálás

1NF-nek megfelel, mert az összes attribútum atomi tulajdonságú.
2NF-nek megfelel, mert minden entitás egy kulccsal rendelkezik és így mindegyik másodlagos attribútum (nem kulcs) teljes mértékben függ a kulcstól
3NF-nek megfelel, az attribútumok között nincs tranzitív függés

Táblatervek

Felhasznalo

felhasznalonev	VARCHAR(50)	A felhasználó felhasználóneve, kulcs
nev	VARCHAR(50)	A felhasználó neve
jelszo	VARCHAR(256)	A felhasználó jelszava

Név: Vrabély Gellért
Neptun kód: RWQQ36

Tag

tag_id	INT	A tag azonosítója, kulcs
nev	VARCHAR(50)	A tag neve
szuletesi_datum	DATE	A tag születési dátuma
poszt	VARCHAR(50)	A tag posztja
felhasznalonev	VARCHAR(50)	Külső kulcs a Felhasznalo táblához
csapat_id	INT	Külső kulcs a Csapat táblához

Tag_allampolgarsag

tag_id	INT	Külső kulcs a Tag táblához, kulcs
allampolgarsag	VARCHAR(50)	A tag állampolgársága
felhasznalonev	VARCHAR(50)	Külső kulcs a Felhasznalo táblához

Csapat

csapat_id	INT	A csapat azonosítója, kulcs
nev	VARCHAR(50)	A csapat neve
varos	VARCHAR(50)	A csapat városa
alapitas_eve	DATE	A csapat alapítási éve
felhasznalonev	VARCHAR(50)	Külső kulcs a Felhasznalo táblához

Név: Vrabély Gellért
Neptun kód: RWQQ36

Merkozes

merkozes_id	INT	A mérkőzés azonosítója, kulcs
eredmeny	VARCHAR(50)	A mérkőzés eredménye
datum	DATE	A mérkőzés dátuma
helyszin	VARCHAR(50)	A mérkőzés helyszíne
nyertes_id	INT	A győztes tag azonosítója, külső kulcs a Csapat táblához
felhasznalonev	VARCHAR(50)	Külső kulcs a Felhasznalo táblához

Merkozik

csapat_id	INT	Külső kulcs a Csapat táblához, kulcs
merkozes_id	INT	Külső kulcs a Merkozes táblához, kulcs

Összetett lekérdezések : CompleyQueryController.java

1. **Kettős állampolgárok:**
   ```
   SELECT nev
   FROM tag
   WHERE tag_id IN
         (SELECT tag_id FROM tag_allampolgarsag GROUP BY tag_id HAVING
         count(allampolgarsag) > 1);
   ```
   A lekérdezés azt gyűjti ki, hogy melyek azok a tagok, akiknek több állampolgárságuk van (legalább 2)
2. **Mérkőzések száma csapatonként:**
   ```
   SELECT csapat.nev AS nev, count(merkozik.merkozes_id) as merkozesek
   FROM merkozes
   INNER JOIN merkozik ON merkozes.merkozes_id = merkozik.merkozes_id
   INNER JOIN csapat ON csapat.csapat_ID = merkozik.csapat_id
   WHERE csapat.nev IN (*táblák*)
   GROUP BY csapat.nev;
   ```

A lekérdezés megadja, hogy az alkalmazásban kiválasztott táblák (ez lesz felsorolva a (*táblák*) helyén) mennyi mérkőzést játszottak

3. **Egy csapat első 3 tagja névsor szerint:**
   SELECT tag.nev
   FROM tag
   WHERE tag.csapat_id = (SELECT csapat_id FROM csapat WHERE nev = csapatnév)
   ORDER BY tag.nev LIMIT 3;
   A lekérdezés megadja, az alkalmazásban kiválasztott csapat (csapatnév) 3 játékosát akik a névsor elején vannak.

4. **Nyerések csapatok szerinti bontásban:**
   SELECT csapat.nev, count(nyertes_id) as nyert
   FROM csapat
   LEFT JOIN merkozes ON csapat.csapat_id = merkozes.nyertes_id
   GROUP BY csapat.nev
   ORDER BY count(nyertes_id) DESC;
   A lekérdezés megadja, a csapatok hány mérkőzést nyertek meg mérkőzés nyerés szerint csökkenő sorrendben

Megvalósítás, funkciók

A projektemet Javában írtam, 17.0.10 verziószámú SDK-t használva. A GUI JavaFX-el van megvalósítva, így ebben a csomagban lévő libraryket használtam fel. Továbbá a MySQL szerverhez való kapcsolódáshoz JDBC-t használtam. A MySQL szervert a Xampp keretrendszerrel értem el és kezeltem.
A projekt packagekbe van rendezve:

- src
  - main
    - java
      - com.teamsportsdb
        - ui
          - ComplexQueryController.java
          - ConnectController.java
          - MainController.java
          - RegistrationController.java
          - UserDashBoardController.java
        - utils
          - DashBoardUtils.java
          - Database.java
          - LoginManager.java
          - SceneManager.java
        - MainApplication.java
    - resources
      - com.teampsortsdb
        - ComplexQuery.fxml
        - Connect.fxml

Név: Vrabély Gellért
Neptun kód: RWQQ36

Main.fxml
Registration.fxml
UserDashboard.fxml

A utils mappában statikus függvények vannak, amik újra és újra fel vannak használva a kód során. Az fxml fájlokban találhatók maguk az objektumok, amik megjelennek és amiket kezel egy felhasználó. Ezekhez tartoznak Controller-ek, ezek vannak a ui mappában, ezek teszik lehetővé, hogy a felhasználó valamely módon bele tudjon szólni a program működésébe.

A program elindításakor meg kell adni az adatbázishoz tartozó felhasználónevet és jelszót, mivel Xampp keretrendszert használtam, nincs jelszó, a felhasználónév: root. Ezután egy üdvözlő üzenet tűnik fel 3 mp-ig, majd bedob minket a Main.fxml-re. Itt tudunk a komplex lekérdezésekhez (ez belépés nélkül elérhető), regisztrációhoz navigálni; vagy belépünk, mint felhasználó, ebben az esetben bedob minket a UserDashboard.fxml-re, ahol tudunk lekérdezni, módosítani, változtatni és törölni adatokat, mindez dinamikusan van megoldva, egyes értékeket kell csak a felhasználónak megadni, a többi kiválaszható legördülő listából. Ha esetleg a regisztrációhoz navigálunk és sikeresen regisztrálunk, akkor is a UserDashboard-hoz irányít minket az alkalmazás, de minden oldalról van lehetőségünk visszalépni a Main.fxml-re.
Ha a komplex lekérdezésekhez navigálunk, 4-féle lekérdezést tudunk futtatni, ezekhez CheckBox, vagy ChoiceBox objektumok vannak felhasználva, hasonlóan, mint a UserDashboard-nál.