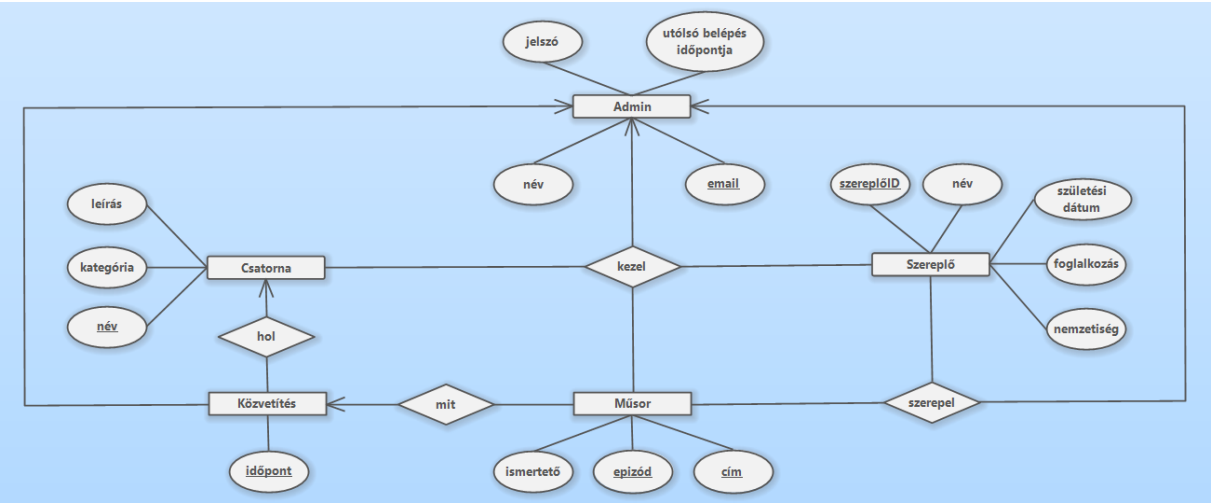
Műsorújság

Specifikáció:
A megvalósítandó alkalmazás az egyes tévécsatornák műsorkínálatában való böngészésre szolgál. Bárki megtekintheti a műsorújságot, de csak a regisztrált adminisztrátorok aktualizálhatják az adatokat bejelentkezés után. Az adminok felvehetnek új csatornákat, műsorokat és szereplőket, a műsorokhoz hozzárendelhetnek szereplőket és a sugárzás/közvetítés időpontját és csatornáit.

Tárolt adatok (nem feltétlen jelentenek önálló táblákat):
- Admin: név, jelszó, e-mail, utolsó belépés időpontja
- Csatorna: név, kategória, leírás
- Műsor: cím, epizód, szereplők, ismertető, időpont és hol sugározzák
- Szereplő: név, születési dátum, nemzetiség, foglalkozás

Relációk az adatok között:
Egy műsort több alkalommal és akár több csatornán is vetíthetnek. Egy szereplő több műsorban is szerepelhet.

E-K diagram



Egyedek és attribútumaik

- **Admin**: név, jelszó, e-mail, utolsó belépés időpontja
- **Csatorna**: név, kategória, leírás
- **Közvetítés**: időpont
- **Műsor**: cím, epizód, ismertető
- **Szereplő**: szereplőID, név, születési dátum, nemzetiség, foglalkozás

Azért választottam a közvetítés egyeddé alakítását, mivel egy műsort egyszerre leadhatnak egy időben, több helyen, illetve ugyanazon a helyen többször is ismételhetik ugyanazt. Lehetett volna egy „közvetíti” kapcsolat is, aminek a közvetítési idő az attribútuma, így viszont nem tudtam volna mesterséges azonosítók bevezetése nélkül megkülönböztetni az ismétléseket, illetve több

csatornán játszódó sportközvetítéseket. Műsor táblába is kerülhetett volna a közvetítés helye, ideje, viszont logikusabbnak tűnt elszeparálni a csatornát, magát a közvetítést és a műsort. Így csökken a redundancia, hiszen a műsor táblában nem lesznek felesleges információk (közvetítés helye, ideje).

2. Kapcsolatok

- **kezel**: 1:N-hez típusú, hiszen egy admin több egyedet is felvihet/módosíthat a kínálatba/kínálatból, viszont egyszerre csak egy admin dolgozhat egy egyeden.
- **szerepel**: N:M-hez típusú, hiszen egy műsorban több szereplő is szokott szerepelni, illetve egy szereplő (jó esetben) több műsorban is szerepel.
- **hol**:
    - 1:N-hez típusú, hiszen egy egy műsort több csatorna is közvetíthet, de egy csatorna egyszerre csak egy műsort közvetíthet.
- **mit:**
    - 1:N-hez típusú, mivel egy műsor egyszerre több közvetítésen keresztül adhatják le, viszont egy közvetítés egyszerre csak egy műsort közvetít.

Leképezés relációsémákká

1. Egyedek és attribútumaik leképezése

   A következő egyedek könnyen leképezhetőek; az egyed nevével felveszek külön relációsémákat és felsorolom az egyed attribútumait.
   - **ADMIN**(e-mail, admin_név, jelszó, utolsó belépés dátuma)
   - **CSATORNA**(csatorna_név, kategória, leírás)
   - **KÖZVETÍTÉS**(időpont, *csatorna_név*, *műsor_cím, epizód*): E-K diagramon külső kulcs nem jelölhető, de csak így teljes egy közvetítési esemény.
   - **SZEREPLŐ**(szereplőID, név, születési dátum, nemzetiség, foglalkozás)
   - **MŰSOR**(műsor_cím, epizód, ismertető)

2. Kapcsolatok és attribútumaik leképezése

   Fel kell tüntetni a saját, attribútumokat, valamint a kapcsolódó egyedek kulcsait.
   - **KEZELTE_CSATORNA**(*e-mail*, *csatorna_név*)
   - **KEZELTE_MŰSOR**(*e-mail*, *műsor_cím, epizód*)
   - **KEZELTE_SZEREPLŐ**(*e-mail*, *szereplőID*)
   - **KEZELTE_KÖZVETÍTÉS**(*e-mail*, *időpont*, *csatorna_név*, *műsor_cím*, *epizód*)
   - **KEZELTE_MŰSOR_SZEREPLŐI**(*e-mail*, *szereplőID*, *műsor_cím*, *epizód*)
   - **MŰSOR_SZEREPLŐI**(*szereplőID*, *műsor_cím, epizód*)
   - **KÖZVETÍTÉS_HELYE**(*csatorna_név*, *időpont*)
   - **KÖZVETÍTETT_MŰSOR**(*műsor_cím, epizód, időpont*)

3. Konszolidálás

Az 1:N-hez kapcsolat általában az N-oldai egyedből leképezett relációsémával vonható össze. A közvetített_műsor azért nem vonható össze a műsorral, mert a műsorban nincs időpont attribútum, így a műsor kulcsai, csak részhalmaza a közvetített_műsor kapcsolat kulcsainak. Tehát, ebben az esetben az 1-oldai egyeddel, azaz a közvetítéssel vonható össze, mivel a közvetített_műsor kulcsai részhalmaza a közvetítés egyed kulcsainak. A közvetítés_helye, viszont beolvasztható a közvetítés (N-oldali) egyeddel.

Továbbiakban felírom a funkcionális függéseket, meghatározom a kulcsokat, megnézem megfelelnek-e a normálformáknak a relációsémák.

Először meg kell nézni a relációs adatbázissémákon belül az attribútumok közötti kapcsolatokat. Ha nem meghatározó / nem biztosít elég egyediséget egy attribútum, akkor nincs tőle függő másik attribútum.

- **ADMIN**(e-mail, admin_név, jelszó, utolsó belépés dátuma)
  - {e-mail}: Egy e-mail-címhez nem tartozhat több személy (ha csak nem együtt használják ugyanazt az e-mail-címet, amit a levelezőprogramok általában nem tartanak számon), így ez a legegyedibb attribútum.
  - {admin_név}: Gyakori nevek miatt nem meghatározó.
  - {jelszó}: Több felhasználó használhatja ugyanazt a jelszót, nem elég meghatározó.
  - {utolsó belépés dátuma}: Van rá esély, hogy több felhasználó egyszerre lép be, akár másodpercre pontosan.
  - {e-mail, …}: Bármely másik attribútummal párosulva is a maradék attribútum halmaz függeni fog ettől a halmaztól, hiszen az e-mailtől is függnek.
  - {admin_név, jelszó}: Lehet két ugyanolyan nevű admin, akik ugyanazzal a jelszóval rendelkeznek.
  - {admin_név, utolsó belépés dátuma}: Arra is van esély, hogy névrokon adminok ugyanakkor lépnek be.
  - {jelszó, utolsó belépés dátuma}: Ugyanazzal a jelszóval is van esély rá, hogy felhasználók egyszerre lépjenek be.

- **CSATORNA**(csatorna_név, kategória, leírás)
  - {csatorna_név}: Ha csak egy régión belüli csatornákról van szó, akkor feltételezem, nem lesznek ugyanolyan nevű csatornák (hiszen a branding része a csatorna neve), így ez lesz a legmeghatározóbb attribútum. Egyébként, régiók között, lehetne mesterséges azonosítóval (csatornaID) biztosítani a nagyobb fokú egyediséget.
  - {kategória}: Tartozhat két ugyanolyan nevű csatorna ugyanazon kategóriához.
  - {leírás}: Ugyanazon kategóriához tartotó csatornáknál előfordulhat megegyező leírás (pl. üres, általános hírcsatornás leírások stb.)
  - {csatorna_név, …}: -||-

    - {kategória, leírás}: Előfordulhatnak csatornák ugyanazzal a leírással, amik ugyanazon kategóriába vannak sorolva.

- **SZEREPLŐ**(szereplőID, szereplő_név, születési dátum, nemzetiség, foglalkozás)
    - {szereplő_név}: Gyakori nevek miatt nem meghatározó.
    - {születési dátum}: Önmagában nem biztosít elég egyediséget.
    - {nemzetiség}: -||-
    - {foglalkozás}: -||-
    - {szereplő_név, születési dátum}: Feltétezhető, hogy a szereplők halmaza így elég egyediséget biztosítana, mivel közszereplőkből, születési dátummal kiegészítve, adott régión belül, kis eséllyel lennének ismétlődések. Itt azonban inkább bevezetek egy mesterséges azonosítót (szereplőID), ami az egyedüli kulcs lesz.
    - {szereplőID, …}: -||-
    - {nemzetiség, foglalkozás}: Főleg ha egy adott régión belül született szereplőkről van szó, gyakori lesz ezekben az attribútumokban az ismétlődés.
    - {szereplő_név, nemzetiség, foglalkozás}: Nagyobb fokú az egyediség, de még mindig gyakran lehetnek ismétlődések.
    - {szereplő_név, születési dátum, nemzetiség, foglalkozás}: Ritka véletlennek hangzik, hogy ennyi attribútuma megegyezne két szereplőnek, de nem kizárható előfordulás ez sem.

- **MŰSOR**(műsor_cím, epizód, ismertető)
    - {műsor_cím}, {epizód}: Önmagában nem elég meghatározó.
    - {ismertető}: Nem biztos, hogy epizód száma, illetve a műsor címe is benn van.
    - {műsor_cím, epizód}: Így lehet tudni, hogy melyik műsor hányadik részéről van szó, tehát egyértelműen meghatározza a műsort.
    - {műsor_cím, epizód, …}: -||-
    - {ismertető, epizód}: Nem feltétlenül adnak ezek annyi információt, hogy a többi attribútumok függjenek tőlük (Mi a címe a műsornak?)

- **KÖZVETÍTÉS**(*csatorna_név*, *műsor_cím, epizód*, időpont)
    - Négynél kevesebb attribútum nem tudja biztosítani a sorok egyediségét.

A kapcsolatokat is megvizsgálom, bár egyikben sincs attribútum, ami ne lenne kulcs.

- **MŰSOR_SZEREPLŐI** (*műsor_cím*, *epizód*, *szereplőID*)
    - {műsor_cím}, {epizód}, {műsor_cím, szereplőID}: Egy műsorban általában több szereplő is szerepel, illetve több epizódban szerepelhetnek ugyanazok a szereplők.
    - {műsor_cím, epizód, szereplőID}: Így lehet biztosítani, a fenti jelenséget.

- **KEZELTE_CSATORNA**(*e-mail*, *csatorna_név*)
- **KEZELTE_MŰSOR**(*e-mail*, *műsor_cím*, *epizód*)
- **KEZELTE_SZEREPLŐ**(*e-mail*, *szereplőID*)
- **KEZELTE_KÖZVETÍTÉS**(*e-mail, időpont*)
- **KEZELTE_ MŰSOR_SZEREPLŐI**(*e-mail*, *műsor_cím*, *epizód*, *szereplőID*)

  Ezek a kapcsolatok, az admin azonosságát köti össze a táblában eszközölt módosításokkal. Megjegyzem, ha nagyobb fokú nyomon követhetőségre lenne igény, lehetne naplózni adatbázis szinten a módosítás típusát (hozzáadás, módosítás, törlés), időbélyeget, illetve, akár a módosítás pontos részleteit (mi volt, mi lett) is, bár így a kapcsolat nem igazán relációs lenne, mivel trigger logika épülne a táblamódosításokra, nem konkrét kapcsolótábla az egyedek közé. Ezt, szekvencia diagrammal lehetne ábrázolni. Végül úgy döntöttem, hogy maradnak a kapcsolótáblák, amiket szintén a program tudna feltölteni.

Funckionális függések

Admin: {e-mail} -> {admin_név, jelszó, utolsó belépés dátuma}

Csatorna: {csatorna_név} -> {kategória, leírás}

Szereplő: {szereplőID} -> {szereplő_név, születési dátum, nemzetiség, foglalkozás}

Műsor: {műsor_cím, epizód} -> {ismertető}

Közvetítés: {csatorna_név, műsor_cím, epizód, időpont} -> teljes rekord

Kapcsolatoknál attribútum nem függ a kulcsoktól, így ezeket nem írtam fel.

Kulcsok

- {e-mail} += {admin_név, jelszó, utolsó belépés dátuma}
- {csatorna_név} += {kategória, leírás}
- {szereplőID} += {szereplő_név, születési dátum, nemzetiség, foglalkozás}
- {műsor_cím, epizód} += {ismertető, kategória, leírás}
- {csatorna_név, műsor_cím, epizód, időpont}

Normálformák

- 1NF: Minden attribútum atomikus (egyszerű) értékű, nincs összetett attribútum, amit külön sémába kellett volna tenni.
- 2NF: Minden kulcs elsődleges attribútum, és ezek halmazaitól (nem részhalmazaitól) függ minden másodlagos attribútum. Tehát minden másodlagos attribútum teljesen függ bármely kulcstól.
    - ADMIN: Mivel az összes attribútum függ az {e-mail} kulcstól, ez a reláció megfelel a 2NF-nek.

    - CSATORNA: Minden attribútum függ a {csatorna_név} kulcstól, így ez is 2NF.
    - SZEREPLŐ: Minden attribútum a {szereplőID}-től függ, tehát ez 2NF.
    - MŰSOR: Minden attribútum a {műsor_cím, epizód} kulcstól függ, így ez is 2NF.
    - KÖVETÍTÉSEK: Egy attribútuma kulcs, a többi külső kulcs. A kulcsok, kompozit kulcsot alkotnak, tehát minden attribútum függ a teljes kulcstól; 2NF teljesül.
    - Kapcsolatok: Nem tartalmaznak nem kulcs attribútumokat, ezért ezek automatikusan megfelelnek a 2NF-nek, hiszen bármely attribútum része az egyes kulcsoknak.
- 3NF: Nincs tranzitív függőség (X -> Y -> T, X -> Z), így minden minden másodlagos attribútum közvetlenül függ bármely kulcstól.
    - ADMIN: Minden attribútum közvetlenül az {e-mail} kulcstól függ, nincs tranzitív függőség, tehát ez 3NF.
    - CSATORNA: Minden attribútum közvetlenül a {csatorna_név} kulcsoktól függ, így ez is 3NF.
    - SZEREPLŐ: Minden attribútum közvetlenül a {szereplőID}-től függ, így 3NF.
    - MŰSOR: Minden attribútum közvetlenül a {csatorna_név, cím} kulcstól függ, nincs tranzitív függőség, tehát 3NF.
    - KÖZVETÍTÉS: Kompozit kulcsot alkotnak az attribútumai, minden tagja egymástól függ; 3NF teljesül.
    - Kapcsolatok: A kapcsolatok tranzitív függőséget sem tartalmaznak, így ezek is megfelelnek a 3NF-nek.
- BCNF: A relációs sémák mindegyike BCNF-ben van, mert nincs olyan reláció, ahol egy nem kulcs attribútum egy részleges kulcstól függne.

Összetett lekérdezések

Statisztikákat mérnek, ezek is a server.js fájlban találhatók.

- Melyik szereplő szerepelt a legtöbb különböző műsorban egy adott csatornán:

  SELECT k.csatorna_nev, sz.szereplo_nev,

  COUNT(DISTINCT m.musor_cim) AS musorok_szama

  FROM szereplo sz

  JOIN musor_szereploi msz ON sz.id = msz.szereplo_id

  JOIN musor m ON msz.musor_cim = m.musor_cim AND msz.epizod = m.epizod

  JOIN kozvetites k ON m.musor_cim = k.musor_cim AND m.epizod = k.epizod

  GROUP BY k.csatorna_nev, sz.szereplo_nev

```
ORDER BY musorok_szama DESC;
```

- Műsoronként megszámolja, hogy hány különböző szereplő szerepelt benne:

```
SELECT m.musor_cim, m.epizod,
COUNT(DISTINCT ms.szereplo_id) AS szereplok_szama
FROM musor m
JOIN musor_szereploi ms
ON m.musor_cim = ms.musor_cim AND m.epizod = ms.epizod
GROUP BY m.musor_cim, m.epizod
ORDER BY szereplok_szama DESC;
```

- Legnépszerűbb epizódok (részlekérdezéssel), avagy legtöbbet közvetített epizódok minden csatornán:

```
SELECT m.musor_cim, m.epizod, COUNT(*) as kozvetitesek_szama
FROM musor m
JOIN kozvetites k ON m.musor_cim = k.musor_cim AND m.epizod = k.epizod
GROUP BY m.musor_cim, m.epizod
HAVING COUNT(*) > (
                SELECT AVG(kozvetites_db)
                FROM (
                SELECT COUNT(*) as kozvetites_db
                FROM kozvetites
                GROUP BY musor_cim, epizod
        ) as atlag_kozvetitesek
)
ORDER BY kozvetitesek_szama DESC;
```

Integritás

- Ha törlődik egy csatorna, vagy egy műsor, akkor a közvetítés táblában releváns rekordok is törlődnek.
- Ha csak egy epizód törlődik, akkor csak annak az epizódnak a közvetítései fognak törlődni.
- Ha törlődik egy szereplő, akkor a szereplései is törlődni fognak.
- Ha egy admin törlődne, akkor a kezelései nem fognak törlődni, mivel ez jelentős információ lehet a jövőben is.

Program

A web alkalmazáshoz (GUI-nak fogom nevezni) Node.js + Express (back-end) és Angular + Bootstrap (front-end) kombókat használtam. Architektúrája MVC felépítésű, a szerver és a kliens függetlenek, API végpontokon keresztül kommunikálnak. Az adatok HTML táblázatokban jelennek meg.

Főbb funkciók

- Közös:
  Read operáció, ~~keresés~~ [részlegesen megvalósított]
- Admin oldal:
  A következő linken keresztül érhető el: apiURL/admin, ahol az apiURL (alapértelmezetten http://localhost:3000/musorujsag) szerver-GUI interfészként szolgál. Elérésekor egy űrlapon kell megadni az email-címet, illetve a jelszót. Ezután, az adatok ellenőrzéséhez, el kell hogy érjük az adatbázis admin tábláját, a következő útvonalon keresztül: ellenőrző metódus -> service -> szerver -> adatbázis -> admin tábla. Sikertelen belépés estén az alkalmazás figyelmeztet, hogy nem lettek jó belépési adatok megadva. Egyéb esetben pedig történik egy lekérés az adatbázis következő tábláira: csatorna, műsor, közvetítések, szereplések, szereplő. Az adatmanipulációk (C, UD) ikonokkal érhetőek el, az adott elem saját sorában.

  További funkciók:
  - Admin session élettartama (sütiben)
  - Kijelentkezés
  - Ugrás látogatói nézetre
  - Statisztikák lekérdezése
  - ~~Felhasználók lekérdezése~~ [részlegesen megvalósított]
  - ~~Naplózás lekérdezése~~ [részlegesen megvalósított]
  - ~~Média tartalmak hozzáadása csatornához, műsorhoz~~ [részlegesen megvalósított]

- Látogató

  Az API URL-jén keresztül érhető el. Eléggé bare-bones, nem jutott idő UX oldali fejlesztésekre.

1. kép: Ilyesmi lehetett volna… https://www.adultswim.com/videos

Könyvtárak/Modulok

- Node.js:

    - Express: Keretrendszer, útvonalkezeléshez.
    - MySQL2: Adatbázis eléréséhez.
    - Body-parser: Lekérések törzsének elemzéséhez.
    - CORS: Frontenddel való kommunikációhoz.
- Angular:
    - Bootstrap: Mobilbarát, illetve reszponzív oldalakhoz hasznos keretrendszer.
    - NgModule: Angular modulok létrehozásához, alkalmazás szerkezetének definiálásához.
    - BrowserModule: Ahhoz kell, hogy az alkalmazás futhasson a böngészőben.
    - CommonModule: Alap direktívák, csővezetékek (pl. NgIf).
    - Injectable: Szolgáltatásokat tesz elérhetővé más osztályok/szolgáltatások számára.
    - HttpClientModule: HTTP kérések végrehajtásához.
    - Router, Routes, AppRoutingModule: Komponensek közötti route-oláshoz.
    - Observable, Map, of: Aszinkron adatfolyamok kezeléséhez, reaktív programozáshoz.
    - catchError: Hibakezeléshez.
    - CanActivate: Route-ok hozzáférhetőségét határozza meg.
    - ViewChild: Template változók komponensekben való eléréséhez.
    - ElementRef: DOM elemek komponensen belüli eléréséhez.
    - FormsModule: Űrlapok kezeléséhez, validálásához.
    - DataService: Szerverlekérések helye.
    - MediaService: Komponensek közötti fájl eléréshez.
    - LatogatoComponent: Főoldal, látogatók számára.
    - AdminComponent: Adminok számára biztosítja GUI-n keresztüli adatbázismanipulációt, külön oldalon.
    - MusorComponent: Adott műsorról ad bővebb információkat, külön oldalon.