

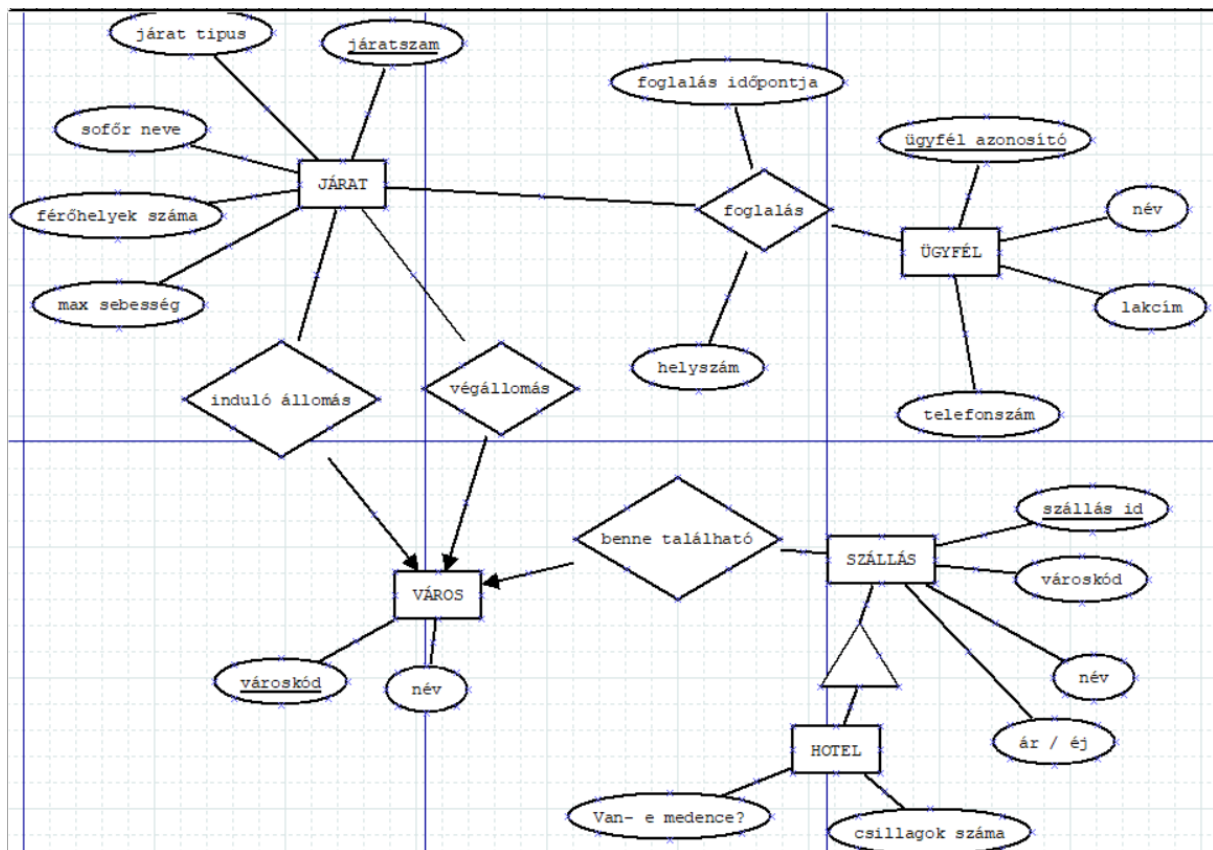
## Busz, Vonat, Repülő

### Jegyfoglalás

Adatbázisok kötelező feladat

A jegyfoglalás webapplikáció járatok, szállások listázására szolgál. Az adatbázisban szereplő (vagy felvitt) ügyfelek foglalhatnak jegyet az elérhető járatokra. Szállásokat kereshetnek az elérhető városokban.

### Egyed-kapcsolat modell



Az adatmodellben 5 darab egyed található:

- Járat (5 attribútum, 3 kapcsolat= 2db 1: N, 1db N:M)
- Ügyfél (4 attribútum, 1 kapcsolat N:M)
- Város (2 attribútum, 3 kapcsolat 3db 1: N)
- Szállás (4 attribútum, 2 kapcsolat 1db 1: N, 1 db specializáló kapcsolat)
- Hotel (speciális egyed, 2 db attribútum, 1 db specializáló kapcsolat)

- Az Ügyfél és a Járat egyedet a foglalás kapcsolat köti össze. Ennek két plusz attribútuma van.
- A Város és Járat egyedet az Induló állomás és végállomás kapcsolat köti össze.
- A Szállás és Város egyedet a benne található kapcsolat köti össze.
- A Hotel egyed az egy speciális Szállás.

## Relációs adatbázisséma

ÜGYFÉL(ügyfél azonosító, név, lakcím, telefonszám)

JÁRAT(járatszám, járat típusa, sofőr neve, férőhelyek száma, max sebesség, *induló állomás.városkód*, *végállomás.városkód*)

VÁROS(városkód, név)

FOGLALÁSOK(járatszám, ügyfélaazonosító, foglalás időpontja, helyszám)

SZÁLLÁS(szállás id, városkód, név, ár/éj)

HOTEL(szállás id, csillagok száma, van-e medence)

## Normalizálás

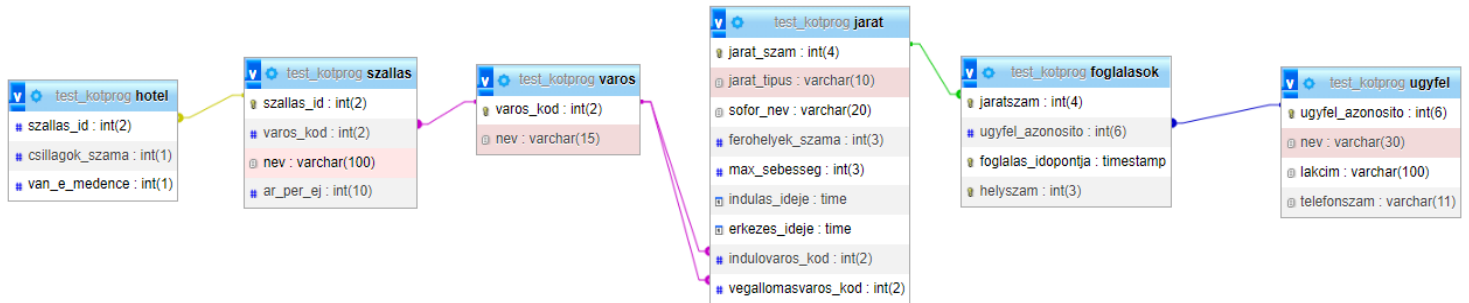
### 2NF:

- Az ÜGYFÉL séma 2NF-ben van, mert egyetlen kulcsa az ügyfél azonosító, és ez ettől minden másodlagos attribútum függ.
- A JÁRAT séma 2NF-ben van, mert egyetlen kulcsa a járatszám, és ez ettől minden másodlagos attribútum függ.
- A VÁROS séma 2NF-ben van, mert egyetlen kulcsa az ügyfél azonosító, és ez ettől minden másodlagos attribútum függ.
- A FOGLALÁSOK séma 2NF-ben van, mert egyetlen másodlagos attribútuma sincs.
- A SZÁLLÁS séma 2NF-ben van, mert a {név, ár/éj} halmaz függ a {szállás id, városkód} kulcsoktól.

### 3NF:

- Az ÜGYFÉL séma 3NF-ben van, mert nincs benne tranzitív függés.
- A JÁRAT séma 3NF-ben van, mert nincs benne tranzitív függés.
- A VÁROS séma 3NF-ben van, mert nincs benne tranzitív függés.
- A FOGLALÁSOK séma 3NF-ben van, mert nincs benne tranzitív függés.
- A SZÁLLÁS séma 3NF-ben van, mert nincs benne tranzitív függés.

## Táblatervek



SQL:

Foglalások:

```
CREATE TABLE `foglalasok` (
  `jaratszam` int(4) NOT NULL,
  `ugyfel_azonosito` int(6) NOT NULL,
  `foglalas_idopontja` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
  `helyszam` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_hungarian_ci;
```

Hotel:

```
CREATE TABLE `hotel` (
  `szallas_id` int(2) NOT NULL,
  `csillagok_szama` int(1) NOT NULL,
  `van_e_medence` int(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_hungarian_ci;
```

Járat:

```
CREATE TABLE `jarat` (
  `jarat_szam` int(4) NOT NULL,
  `jarat_tipus` varchar(10) COLLATE utf8mb4_hungarian_ci NOT NULL,
  `sofor_nev` varchar(20) COLLATE utf8mb4_hungarian_ci NOT NULL,
  `ferohelyek_szama` int(3) NOT NULL,
  `max_sebesseg` int(3) NOT NULL,
  `indulas_ideje` time DEFAULT NULL,
  `erkezes_ideje` time DEFAULT NULL,
  `indulovaras_kod` int(2) NOT NULL,
  `vegallomasvaros_kod` int(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_hungarian_ci;
```

Szállás:

```
CREATE TABLE `szallas` (  
  `szallas_id` int(2) NOT NULL,  
  `varos_kod` int(2) NOT NULL,  
  `nev` varchar(100) COLLATE utf8mb4_hungarian_ci NOT NULL,  
  `ar_per_ej` int(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_hungarian_ci;
```

Ügyfél:

```
CREATE TABLE `ugyfel` (  
  `ugyfelazonosito` int(6) NOT NULL,  
  `nev` varchar(30) COLLATE utf8mb4_hungarian_ci NOT NULL,  
  `lakcim` varchar(100) COLLATE utf8mb4_hungarian_ci NOT NULL,  
  `telefonszam` varchar(11) COLLATE utf8mb4_hungarian_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_hungarian_ci;
```

Város:

```
CREATE TABLE `varos` (  
  `varos_kod` int(2) NOT NULL,  
  `nev` varchar(15) COLLATE utf8mb4_hungarian_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_hungarian_ci;
```

Kulcsok:

```
--  
-- A tábla indexei `foglalasok`  
--  
ALTER TABLE `foglalasok`  
  ADD PRIMARY KEY (`jaratszam`,`foglalas_idopontja`,`helyszam`),  
  ADD KEY `ugyfel` (`ugyfelazonosito`);  
  
--  
-- A tábla indexei `hotel`  
--  
ALTER TABLE `hotel`  
  ADD KEY `szallas_id` (`szallas_id`);  
  
--  
-- A tábla indexei `jarat`  
--  
ALTER TABLE `jarat`  
  ADD PRIMARY KEY (`jarat_szam`),  
  ADD KEY `erkezo_kulso_kulcs` (`vegallomasvaros_kod`),  
  ADD KEY `indulo_kulso_kulcs` (`indulovaros_kod`);  
  
--  
-- A tábla indexei `szallas`  
--  
ALTER TABLE `szallas`  
  ADD PRIMARY KEY (`szallas_id`),  
  ADD KEY `kulso_kulcs_varos` (`varos_kod`);  
  
--  
-- A tábla indexei `ugyfel`  
--  
ALTER TABLE `ugyfel`  
  ADD PRIMARY KEY (`ugyfelazonosito`);  
  
--  
-- A tábla indexei `varos`  
--  
ALTER TABLE `varos`  
  ADD PRIMARY KEY (`varos_kod`);
```

```
-- AUTO_INCREMENT a táblához `szallas`
--
ALTER TABLE `szallas`
  MODIFY `szallas_id` int(2) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=38;

-- AUTO_INCREMENT a táblához `ugyfel`
--
ALTER TABLE `ugyfel`
  MODIFY `ugyfelazonosito` int(6) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=39;

-- AUTO_INCREMENT a táblához `varos`
--
ALTER TABLE `varos`
  MODIFY `varos_kod` int(2) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

-- Megkötések a kiírt táblákhoz
--
-- Megkötések a táblához `foglalasok`
--
ALTER TABLE `foglalasok`
  ADD CONSTRAINT `foglalasok_ibfk_1` FOREIGN KEY (`jaratszam`) REFERENCES `jarat` (`jarat_szam`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `ugyfel` FOREIGN KEY (`ugyfelazonosito`) REFERENCES `ugyfel` (`ugyfelazonosito`) ON DELETE CASCADE;

-- Megkötések a táblához `hotel`
--
ALTER TABLE `hotel`
  ADD CONSTRAINT `hotel_ibfk_1` FOREIGN KEY (`szallas_id`) REFERENCES `szallas` (`szallas_id`) ON DELETE CASCADE;

-- Megkötések a táblához `jarat`
--
ALTER TABLE `jarat`
  ADD CONSTRAINT `erkezo_kulso_kulcs` FOREIGN KEY (`vegallomasvaros_kod`) REFERENCES `varos` (`varos_kod`) ON DELETE CASCADE,
  ADD CONSTRAINT `indulo_kulso_kulcs` FOREIGN KEY (`indulovaros_kod`) REFERENCES `varos` (`varos_kod`) ON DELETE CASCADE;

-- Megkötések a táblához `szallas`
--
ALTER TABLE `szallas`
  ADD CONSTRAINT `kulso_kulcs_varos` FOREIGN KEY (`varos_kod`) REFERENCES `varos` (`varos_kod`) ON DELETE CASCADE;
```

## Összetett lekérdezések

### 1. lekérdezés:

A szállás oldalon ki lehet választani, hogy apartmant vagy hotelt keresünk. Mivel csak egy szállás tábla van, és annak egy hotel speciális egyede, így a hotel táblában nem tárolom el csak azoknak a szállásoknak az azonosítóját, amik hotelek is egyben. Ezek mellett még tárolom a hotelekről a csillagok számát és azt, hogy van-e medence. A Szállások oldalon ezzel a lekérdezéssel tudom visszaadni azokat a szállásokat, amik hotelek, vagy nem.

```
1 SELECT *
2 FROM szallas, hotel
3 WHERE szallas.szallas_id NOT IN (SELECT hotel.szallas_id FROM hotel) AND szallas.varos_kod = 2
4 GROUP BY szallas.szallas_id
```

(Most a kettes város kóddal rendelkező városban keresi csak az apartmanokat)

2. lekérdezés:

Ennek a lekérdezés segítségével vissza tudom adni azt a járatot amire a legtöbb ember foglalt jegyet. Majd ennek a járatnak az adatait kiíratom a Foglalások oldalán.

```
1 SELECT jarat.jarat_szam, jarat.jarat_tipus, jarat.sofor_nev, jarat.ferohelyek_szama, jarat.max_sebesseg, jarat.indulas_ideje, jarat.erkezes_ideje, jarat.indulovaros_kod,
   jarat.vegallomasvaros_kod, COUNT(*) as darab
2 FROM foglalasok, jarat
3 WHERE foglalasok.jaratszam = jarat.jarat_szam
4 GROUP BY jarat.jarat_szam
5 ORDER BY darab DESC
6 LIMIT 1
```

3. lekérdezés:

Ez a lekérdezés megnézi, hogy melyik városból indul a legtöbb járat.

```
1 SELECT varos.varos_kod, varos.nev, COUNT(*) as darab
2 FROM varos, jarat
3 WHERE varos.varos_kod = jarat.indulovaros_kod
4 GROUP BY varos.varos_kod
5 ORDER BY darab DESC
6 LIMIT 1
```

4. lekérdezés:

Ezzel meg tudom nézni, hogy specifikus városból mely járatok indulnak. Ezt a Járatok oldalán használom.

```
1 SELECT jarat_szam, jarat_tipus, sofor_nev, ferohelyek_szama, max_sebesseg, indulas_ideje, erkezes_ideje, indulovaros_kod, vegallomasvaros_kod
2 FROM jarat, varos
3 WHERE varos.varos_kod = jarat.indulovaros_kod
4 AND varos.varos_kod = 1
```

5. lekérdezés:

Ezzel meg tudom nézni, hogy egy adott járaton mely helyek foglaltak, így mikor új foglalás érkezik az adott járatra, azt a helyet nem osztom ki addig amíg foglalt.

```
1 SELECT helyszam
2 FROM foglalasok, jarat
3 WHERE jarat.jarat_szam = foglalasok.jaratszam AND
4 jaratszam = 1001
```

## Megvalósítás

A program megvalósításához a Java nyelvet használtam (JDK 17). A JetBrains IDE-jét használtam az IDEA Intellij-t. Spring boot volt az adatbázis kapcsolatot segítő eszköz. Az adatbázis és a program összekötését a DAO osztályokban valósítottam meg, itt a Spring Boot-ba beépített JdbcDaoSupport volt segítségemre. Az SQL kódokat nem generálta le előre, így a programba nekem kellett felvinni a megfelelő SQL kódokat. A megjelenítést HTML+ CSS segítségével oldottam meg. A programot és a HTML oldalakat a Thymeleaf sablonmotor

segítségével kötöttem össze. Néhány formázást online Bootstrap css segítségével oldottam meg.

## Funkciók

Főoldal:

A főoldalon csak egy egyszerű üdvözlő szöveg van. DE a „weboldalon” szövegre kattintva az admin oldalra jutunk. Az admin oldalon vannak azok a dolgok, amik egy alap felhasználó nem férhet hozzá. Pl.: Járatok hozzáadása, törlése...

Ügyfél oldal:

Az ügyfél oldalon tudunk hozzáadni egy ügyfelet az adatbázishoz. Mikor hozzáadjuk az ügyfelet, akkor megjelennek az adatai a lentebb található táblázatban. Ebben a táblázatban tudjuk törölni az adott felhasználót a „Törlés” gombbal, a „Szerkesztés” gombbal meg az adott felhasználó adatai belekerülnek a szerkesztés oldalon található form-ba. Itt tudjuk szerkeszteni az ügyfelünket.

Járatok oldal:

A járatok oldalon ki tudjuk listázni az adott városból induló járatokat. A válassz egy várost legördülő menüben ki tudjuk választani az adatbázisban szereplő városokat. (Ha az admin új várost ad az adatbázishoz, akkor ez a lista is frissül). Egy várost kiválasztva majd az „Elküld” gombra kattintva, megjelennek az adott városból induló járatok adatai. A „Legtöbb járat innen indul:” szöveg alatt annak a városnak a neve jelenik meg amiből a legtöbb járat indul. (Ha új járatokat adunk hozzá az adatbázishoz, ami más városból indul akkor mindig az jelenik meg ahonnan a legtöbb indul).

Foglalások oldal:

A foglalások oldalon két legördülő listát látunk. Az egyikben az adatbázisban szereplő ügyfelek vannak, a másikban meg az adatbázisban szereplő járatok. A listákból kiválasztva az adott ügyfelet és járatot, majd megnyomva az „Elküld” gombot tudunk foglalást leadni. Ez megjelenik a lentebb található listában. Ha egy ügyfél több járatot is foglalt akkor a foglalásai egymás alatt jelennek meg, név szerint csoportosítva. Az ügyfél tudja törölni a foglalását a foglalása sorában levő „Törlés” gombbal. (Az oldalon szintén frissülnek a lenyíló fülek az adatbázis frissítése esetén)

Szállások oldal:

A szállások oldalon a „Szállások listázása” szöveg alatt ki tudjuk választani azt a várost, ahol ki szeretnénk listázni a szállásokat. A lenyíló fül alatt rádió gombokkal ki tudjuk választani, hogy apartmant vagy hotelt akarunk keresni az adott városban. Ha az apartmant választjuk akkor az adott városból kilistázza csak az apartmanokat. (Azokat a szállásokat, amik hotelek is egyben, azokat nem). Ha a hotel rádió gombot válasszuk akkor az alatta található lenyíló menüben ki kell választanunk, hogy hány csillagos hotelt keresünk, majd alatt a rádió gombokkal meg kell adnunk, hogy olyan szállást keressen, ahol van medence. Ha találat van akkor a jobbra található táblázatban megjelennek a szállás/szállások adatai. Ha nincs találat,



akkor nem jelenik meg semmi. (Mivel a hotelek keresésénél nagyon sok kombináció lehet így pár példa: Gyula, Hotel, csillagok száma=4, van e medence = van; Kecskemét, Hotel, csillagok száma=2, van e medence= nincs; Barcelona, Hotel, csillagok száma = 5, van e medence = van) Ha apartmant keresünk akkor nem kell ezeket a dolgokat elvégeznünk, csak kiválasztani a várost, majd az „Elküld” gomb megnyomása után kilistázza a városban található apartmanokat.

Admin oldal:

Mint már említettem a főoldalon a „weboldalon” szövegre kattintva érhetjük el.

Funkciók:

- Járat törlés: Járatszám alapján tudjuk törölni a járatokat.
- Járat hozzáadás: Meg kell adnunk a feltüntetett adatokat:
  - o Járat szám: int, az adott járat száma, egyedi, 1-9999 között
  - o járat típus: string
  - o sofőr neve: string
  - o férőhelyek száma: int
  - o max sebesség: int
  - o indulás ideje: time, formátum: 00:00:00
  - o érkezés ideje: time, formátum: 00:00:00
  - o induló város: legördülő lista
  - o végállomás: legördülő lista
- Járat szerkesztése:
  - o járat szám: legördülő lista
  - o új járat típus: string
  - o új sofőr neve: string
  - o új férőhelyek száma: int
  - o új max sebesség: int
  - o új indulás ideje: time, formátum: 00:00:00
  - o új érkezés ideje: time, formátum: 00:00:00
  - o új induló város: legördülő lista
  - o új végállomás: legördülő lista
- Foglалás szerkesztése:
  - o ügyfél: lista
  - o foglalás időpontja: mikor történt a foglalás (ez a foglalások oldalon látszódik, ugyan abban a formátumban):
  - o volt helyszám (ez a foglalások oldalon látszódik): int
  - o új helyszám: int
- Szállás hozzáadása:
  - o város: mely városban található: lista
  - o név: string
  - o ár: int
- Szállás törlése:



- o a szállás neve alapján tudjuk törölni a szállást
  
- Szállás frissítése:
  - o szállás név: melyik szállást akarjuk frissíteni: lista
  - o új név: string
  - o új ár: int
- Hotelek listázása:
  - o külön kilistáztam az adatbázisban található olyan szállásokat, amik hotelek is egyben, ha véletlen a sok lehetséges kombináció miatt nem lenne találat a Szállások oldalon, a hotelek keresésénél.
- Hotel hozzáadása:
  - o név: azok a szállások, amik nem hotelek: lista
  - o van e medence: 0-1 között, 0 ha nincs, 1 ha van: int
  - o csillagok száma: 1-5 között, int
- Hotel törlése:
  - o szállás törlése a hotelek közül név alapján: lista
- Hotelek szerkesztése:
  - o hotel neve: lista
  - o van e medence (új): 0-1 között, 0 ha nincs, 1 ha van: int
  - o csillagok száma (új): 1-5 között, int
- Város hozzáadása:
  - o név: string
- Város törlése:
  - o város törlése név alapján: lista
- Város szerkesztése:
  - o melyik város, név szerint: lista
  - o város új neve: string

A külső kulcs kapcsolatok miatt egyes táblák módosítása befolyásolnak más táblákat is. PL.: Ha törölünk egy járatot, akkor a járatra érkezett foglalások is törlődnek. Vagy ha törölünk egy várost akkor a oda érkező és onnan induló járatok is törlődnek. Tábla Update-nál is működnek ezek a funkciók.