

Repülőjegy foglalás

Tagok:

Bata András
Kalmár György
Kószó Áron

Csütörtök, 10-12

Tartalomjegyzék

1. Szöveges feladateleírás, követelmény katalógus	2
2. Adatbázis	3
2.1 Egyed-kapcsolat diagram	3
2.2 Normalizálás	4
2.3 Adattáblák	6
3. Adatfolyam diagram	14
4. Egyed-esemény mátrix	22
5. Szerep-funkció mátrix	23

A célunk egy repülőjegyek foglalását lehetővé tévő rendszer megalkotása, webes környezetben. Rendszerünk tervezése során megvizsgáltunk már meglévő rendszereket, mint például: mortons.hu ; repjegy.hu . Mivel ezen szolgáltatások nagyméretű adatállomány gyors, biztonságos, dinamikus hozzáférhetőségét nélkülözhetetlennek tartják, így elengedhetetlen egy ezen szolgáltatásokat kiszolgálni képes adatbázis kezelő rendszer használata, amely esetünkben az Oracle lesz. A munkát három fő szakaszban szeretnénk elvégezni. Az első részben kihasználva az SSADM technikák némelyikét, megtervezzük munkák menetét és rendszerünk egészét. Utána elkészítjük a megvalósítás alappilléreül szolgáló adatbázist, majd végül a rendszer képességeit kihasználni kívánó személyek (továbbiakban felhasználók) számára megfelelő kezelői felületet teremtve rendelkezésre bocsájtjuk a kész projektet.

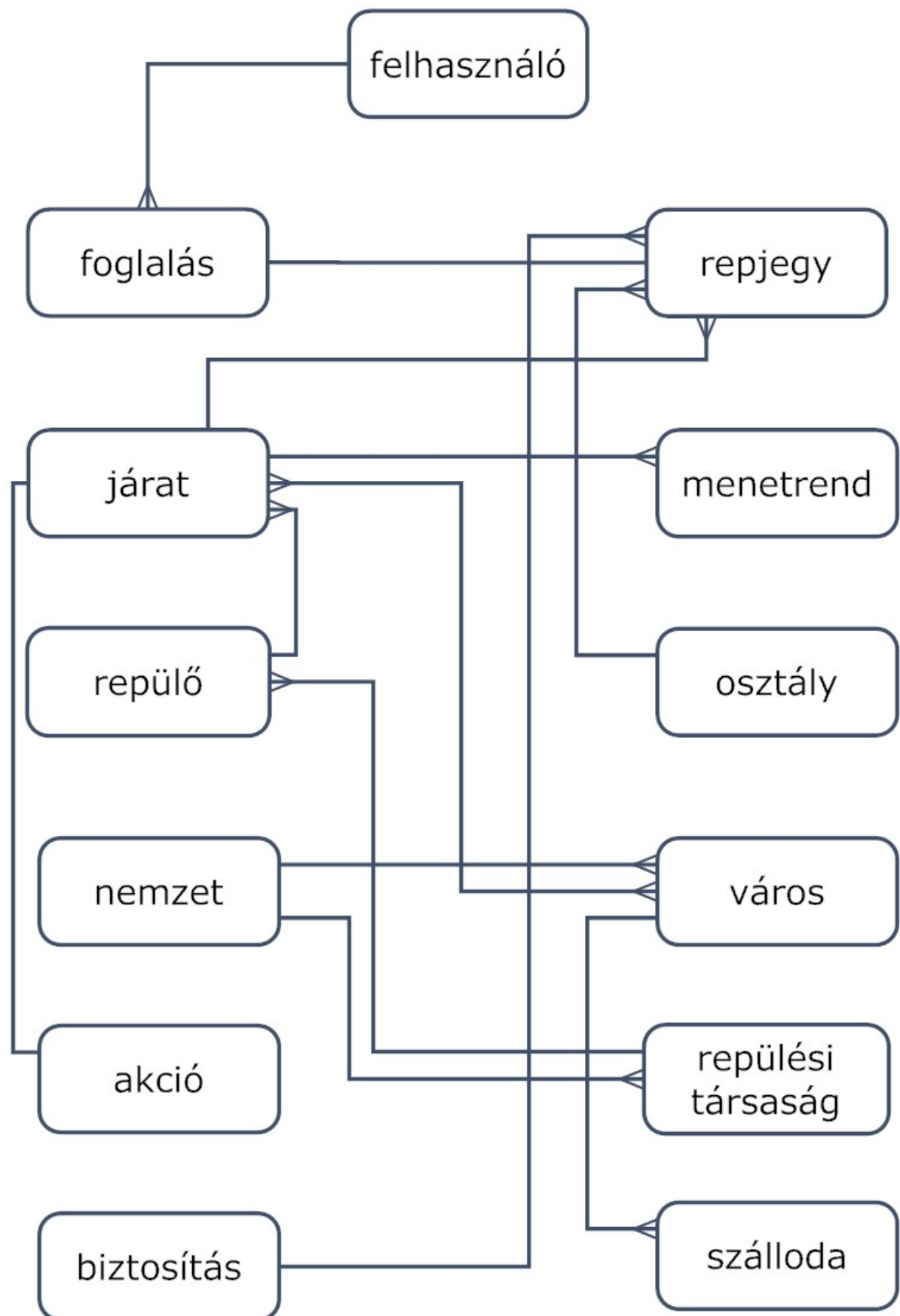
A rendszerrel szemben az alábbi általános követelményeket állítjuk fel:

- repülő járatok, légitársaságok
- menetrend
- átszállások, csatlakozások kezelése
- repülőjárat kereső (dátum alapján, csatlakozások figyelembe vételével)
- legnépszerűbb járatok
- árajánlatok
- éves kimutatás készítése
- Felnőtt/gyerek utaztatás
- Szűrő feltételek (aznapi átszállás, legrövidebb idő, minimális átszállások száma)
- Repülő típusok
- Repülőjegy foglalás (mikor, melyik járat, étkezés van-e?, van-e még hely?, ülőhely kiválasztás), költség kiszámítása
- Biztosítók és biztosítási csomagok kezelése
- A városban található szállodák
- Repülőjegy cseréje

Fejlesztők által támasztott nem funkcionális követelmények:

- gyors, dinamikus, megbízható rendszer
- sok felhasználó egy időben történő kiszolgálása

Adatbázis:



A leképezett sémák:

Felhasználó (**felhasználónév** , jelszó, utazási okmány száma, neve, szül. dátum, bankkártyaszám, lakcím, telefonszám)

Foglalás(**felhasználónév**, **repjegy id**)

Repjegy(**repjegy id**, *járat id*, *osztály száma*, *biztosítás id*)

Járat(**járat id**, *honnan id*, *hova id*, *repülő id*)

Menetrend(**járat id**, **mikor indul**, mikor érkezik)

Repülő(**repülő id**, típus, ülőhely, *társaság id*)

Repülési társaság(**társaság id**, neve, *nemzet id*)

Nemzet(**nemzet id** , országnév)

Város(**város id**, városnév, *nemzet id*)

Akciók(**járat id**, **újár**)

Osztály(**szám** , étkezés van-e?, relaxációs lehetőség?, extra bőrönd?, internet?)

Biztosítás(**biztosítás id**, biztosító neve, poggyászbiztosítás?, életbiztosítás?, különös esetre?)

Szálloda(**város id**, **neve**, leírás)

1NF:

A felhasználó séma lakcím adattagja összetett attribútum, amely állhat város, irányítószám, utca, stb... elemekből, így ezen séma nincs első normálformában. A megoldás egy:

Felhasználó (**felhasználónév** , jelszó, utazási okmány száma, neve, szül. dátum, bankkártyaszám, telefonszám)

Lakcím(**felhasználónév**, ország, város, irányítószám, utca, házszám)

módosítás lenne, azonban esetünkben nem indokolt a módosítás, hiszen csak szöveggént, esetleges postázás céljából, szeretnénk eltávolítani a lakcímet, nem terveztünk a lakcímre vonatkozó szűréseket, ami indokolná az új séma bevezetését.

2NF:

Második normálforma esetén olyan sémákra kell odafigyelni, ahol a kulcs többelemű, tehát esetünkben a Menetrend és a Szálloda sémák érdekesek.

A Menetrend esetén, a "mikor érkezik" adattag esetén kell megvizsgálni, hogy teljesen függ-e a kulcs minden elemétől. Beláthatjuk, hogy igen, hiszen egy időben több járat is indulhat, így az érkezési idő nyilván függ attól, hogy melyik járat, és mikor indult.

A másik vizsgálandó alany a Szálloda. Egy adott szálloda leírása természetesen függ attól, hogy melyik város melyik szállodájáról van szó. Vannak nemzetközi cégek, amelyek minden nagy városban létesítenek szállodát, és nyilván minden szállodájukhoz külön leírás tartozik, mivel más országok, ők is alkalmazkodnak. Tehát ez a séma sem sérti a második normálforma kritériumait.

3NF:

Tranzitív függés keresése a cél. A Felhasználó sémában elő is fordul is rögtön egy. Egy felhasználóhoz egy ember tartozik, akit nyilvánvalóan az okmányának száma egyértelműen azonosít. Tehát a:

Felhasználó (**felhasználónév** , jelszó, *utazási okmány száma*)

Személy(**utazási okmány száma**, neve, szül. dátum, bankkártyaszám, lakcím, telefonszám)

módosítás megállná a helyét, és helyes is lenne. Azonban számunkra ismételten nem fontos ezen változás , hiszen mi a felhasználót, mint embert kezeljük rendszeren belül, nem szeretnénk külön azzal foglalkozni, hogy felhasználókat ilyen formában tároljunk.

Redundanciát egyébként sem okozna, hiszen egy felhasználóhoz egy személy tartozna.

A Repülő sémában az ülőhely talán csak a repülőgép típusától függ, azonban valamilyen okból átalakíthatták a gépet, és kevesebb ülőhely áll rendelkezésre ezután. Ezért ezt is letároljuk mindenesetre.

Egyéb tranzitív függést nem fedeztünk fel a sémákban, így a kezdeti sémák változtatás nélkül képezik a létrehozandó adatbázis alapját.

FELHASZNÁLÓ:

3NF

Felhasználó

username

pass

utazási okmány száma

neve

szül. dátum

bankkártyaszám

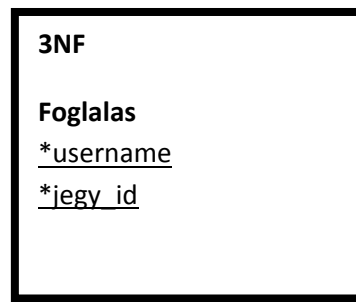
lakcím

telefonszám

Felhasználó		
<u>username</u>	VARCHAR2(16)	A felhasználónév.
pass	VARCHAR2(25)	A jelszó.
okmanyszam	VARCHAR2(45)	Utazási okmányának száma.
neve	VARCHAR2(60)	A felhasználó neve.
szuletett	DATE	A felhasználó születési dátuma.
bankkartyaszam	NUMBER(16)	Bankkártyájának a száma.
lakcim	VARCHAR2(60)	A felhasználó lakcíme.
telefonszam	NUMBER(13)	A felhasználó telefonszáma.

```
CREATE TABLE felhasznalo (  
  username      VARCHAR2(16) primary key not null ,  
  pass          VARCHAR2(25)      ,  
  okmanyszam    VARCHAR2(45)      ,  
  neve          VARCHAR2(60)      ,  
  szuletett     DATE              ,  
  bankkartyaszam NUMBER(16)      ,  
  lakcim        VARCHAR2(60)      ,  
  telefonszam   NUMBER(13)  
);
```

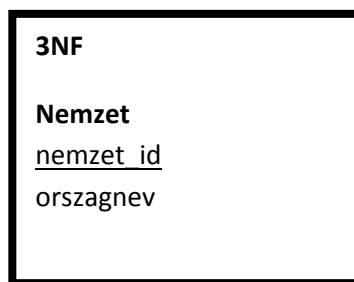
FOGLALÁS:



Foglalas		
<u>*username</u>	VARCHAR2(16)	Egy felhasználóra mutat.
<u>*jegy_id</u>	NUMBER(9)	Egy adott jegyre mutat.

```
CREATE TABLE foglalas (  
  username    VARCHAR2(16)          ,  
  jegy_id     NUMBER(9)             ,  
  PRIMARY KEY (username, jegy_id)   ,  
  FOREIGN KEY (username) REFERENCES felhasználó(username) ON DELETE CASCADE ,  
  FOREIGN KEY (jegy_id)   REFERENCES jegy(jegy_id)         ON DELETE CASCADE  
);
```

NEMZET:



Nemzet		
<u>nemzet_id</u>	NUMBER(3)	Nemzetazonosító.
orszagnev	VARCHAR2(30)	Az ország neve.

```
CREATE TABLE nemzet (  
  nemzet_id   NUMBER(3)   primary key not null ,  
  orszagnev   VARCHAR2(30)  
);
```


REPÜLÉSI TÁRSASÁG:

3NF
Tarsasag
<u>tarsasag_id</u>
tarsasagnev
*nemzet_id

Tarsasag		
<u>tarsasag_id</u>	NUMBER(5)	A repülési társaság azonosítására szolgál.
tarsasag_nev	VARCHAR2(30)	A repülési társaság neve.
*nemzet_id	NUMBER(3)	Egy nemzetre mutat, kiolvasható az ország neve.

```
CREATE TABLE tarsasag (
  tarsasag_id NUMBER(5) primary key not null ,
  tarsasag_nev VARCHAR2(30) ,
  nemzet_id NUMBER(3) ,
  FOREIGN KEY (nemzet_id) REFERENCES nemzet(nemzet_id) ON DELETE CASCADE
);
```

VÁROS:

3NF
Varos
<u>varos_id</u>
varosnev
*nemzet_id

Varos		
<u>varos_id</u>	NUMBER(3)	Egy város azonosítására szolgál.
varosnev	VARCHAR2(30)	A város neve.
*nemzet_id	NUMBER(3)	Egy nemzetre mutat, kiolvasható az ország neve.

```
CREATE TABLE varos (
  varos_id NUMBER(3) primary key not null ,
  varosnev VARCHAR2(30) ,
  nemzet_id NUMBER(3) ,
  FOREIGN KEY (nemzet_id) REFERENCES nemzet(nemzet_id) ON DELETE CASCADE
);
```

REPÜLŐ:

3NF
Repulo
<u>repulo_id</u>
tipus
ulohely
*tarsasag_id

Repulo		
<u>repulo_id</u>	NUMBER(6)	A repülőgép azonosítására szolgál.
tipus	VARCHAR2(50)	A repülőgép típusa.
ulohely	NUMBER(3)	A repülőgép kapacitása.
*tarsasag_id	NUMBER(5)	A repülőgépet birtokló társaság.

```
CREATE TABLE repulo (
  repulo_id    NUMBER(6) primary key not null    ,
  tipus        VARCHAR2(50)                      ,
  ulohely      NUMBER(3)                          ,
  tarsasag_id  NUMBER(5)                          ,
  FOREIGN KEY (tarsasag_id) REFERENCES tarsasag(tarsasag_id) ON DELETE CASCADE
);
```

JÁRAT:

3NF
Jarat
<u>jarat_id</u>
*honnan
*hova
*repulo_id

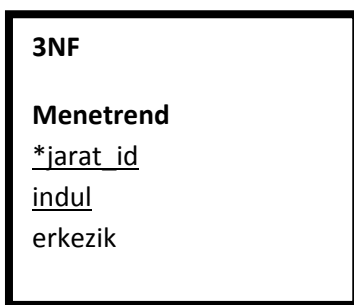
Jarat		
<u>jarat_id</u>	NUMBER(6)	Járat azonosító.
*honnan	NUMBER(3)	Honnan indul a járat (egy városra mutat).
*hova	NUMBER(3)	Hova érkezik a járat (egy városra mutat).
*repulo_id	NUMBER(6)	Mely repülő megy, egy adott repülőre mutat.

```

CREATE TABLE jarat (
  jarat_id      NUMBER(6) primary key not null ,
  honnan        NUMBER(3)                    ,
  hova          NUMBER(3)                    ,
  repulo_id     NUMBER(6)                    ,
  FOREIGN KEY (repulo_id) REFERENCES repulo(repulo_id) ON DELETE CASCADE ,
  FOREIGN KEY (honnan) REFERENCES varos(varos_id) ON DELETE CASCADE ,
  FOREIGN KEY (hova) REFERENCES varos(varos_id) ON DELETE CASCADE
);

```

MENETREND:



Menetrend		
<u>*jarat_id</u>	NUMBER(6)	Melyik járatról van szó.
<u>indul</u>	DATE	Mikor indul..
erkezik	DATE	És mikor érkezik.

```

CREATE TABLE menetrend (
  jarat_id      NUMBER(6) ,
  indul         DATE      ,
  erkezik       DATE      ,
  PRIMARY KEY (jarat_id, indul) ,
  FOREIGN KEY (jarat_id) REFERENCES jarat(jarat_id) ON DELETE CASCADE
);

```

OSZTÁLY:

3NF

Osztaly

szama

etkezes

relax

extra_borond

internet

Osztaly		
<u>szama</u>	NUMBER(1)	Az osztály számát azonosítja (pl: 1 = első osztály)
etkezes	NUMBER(1)	Van-e etkezés?
relax	NUMBER(1)	Van-e pihenési lehetőség?
extra_borond	NUMBER(1)	Vihet-e extra bőröndöt?
internet	NUMBER(1)	Tud-e internetre csatlakozni?

CREATE TABLE osztaly (

szama NUMBER(1) primary key not null ,

etkezes NUMBER(1) ,

relax NUMBER(1) ,

extra_borond NUMBER(1) ,

internet NUMBER(1)

);

SZÁLLODA:

3NF

Szalloda

*varos_id

neve

leiras

Szalloda		
* <u>varos_id</u>	NUMBER(3)	Melyik városban található?
<u>neve</u>	VARCHAR2(30)	Mi a szálloda neve?
leiras	VARCHAR2(4000)	Leírás a szállodáról.

CREATE TABLE szalloda (

varos_id NUMBER(3) ,

neve VARCHAR2(30) ,

leiras VARCHAR2(4000) ,

PRIMARY KEY (varos_id , neve) ,

FOREIGN KEY (varos_id) REFERENCES varos(varos_id) ON DELETE CASCADE

);

BIZTOSÍTÁS:

3NF

Biztositas

bizt_id

utlemondas

poggyasz

arvisszaad

Biztositas		
<u>bizt_id</u>	NUMBER(3)	Biztosítás-csomag azonosítója.
utlemondas	NUMBER(1)	Járat törlésekor biztosítanak-e másik járatot?
poggyasz	NUMBER(1)	Van-e biztosítás a csomagokra?
arvisszaad	NUMBER(1)	Járatkimaradáskor visszatérítik-e a jegy árát?

```
CREATE TABLE biztositas (
  bizt_id      NUMBER(3) primary key not null ,
  utlemondas   NUMBER(1)          ,
  poggyasz     NUMBER(1)          ,
  arvisszaad   NUMBER(1)
);
```

AKCIÓ:

3NF

Akcio

*jarat_id

ujar

Akcio		
* <u>jarat_id</u>	NUMBER(6)	Melyik járatra érvényes az akció?
<u>ujar</u>	NUMBER(6)	Mi az új ár?

```
CREATE TABLE akcio (
  jarat_id     NUMBER(6)          ,
  ujar         NUMBER(6)          ,
  PRIMARY KEY (jarat_id , ujar)    ,
  FOREIGN KEY (jarat_id) REFERENCES jarat(jarat_id) ON DELETE CASCADE
);
```

JEGY:

3NF

Jegy

jegy_id

*jarat_id

*osztaly_id

*birt_id

Jegy		
<u>jegy_id</u>	NUMBER(9)	Jegy azonosítására szolgál.
*jarat_id	NUMBER(6)	Melyik járáshoz tartozik az adott jegy.
*osztaly_id	NUMBER(1)	Melyik osztályra szól a jegy.
*birt_id	NUMBER(3)	Milyen biztosítást választottak a jegyhez.

CREATE TABLE jegy (

 jegy_id NUMBER(9) primary key not null ,

 jarat_id NUMBER(6) ,

 osztaly_id NUMBER(1) ,

 birt_id NUMBER(3) ,

 FOREIGN KEY (jarat_id) REFERENCES jarat (jarat_id) ON DELETE CASCADE ,

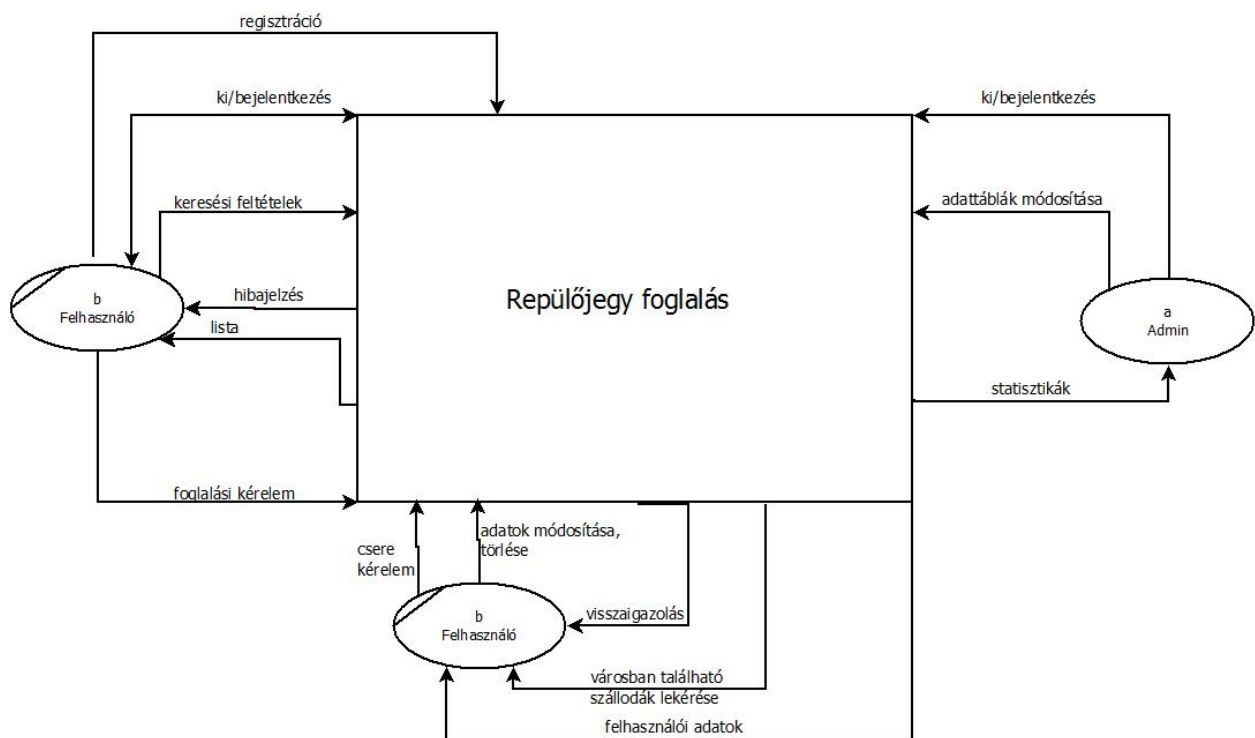
 FOREIGN KEY (osztaly_id) REFERENCES osztaly(szama) ON DELETE CASCADE ,

 FOREIGN KEY (birt_id) REFERENCES biztositas(birt_id) ON DELETE CASCADE

);

Adatfolyam diagram:

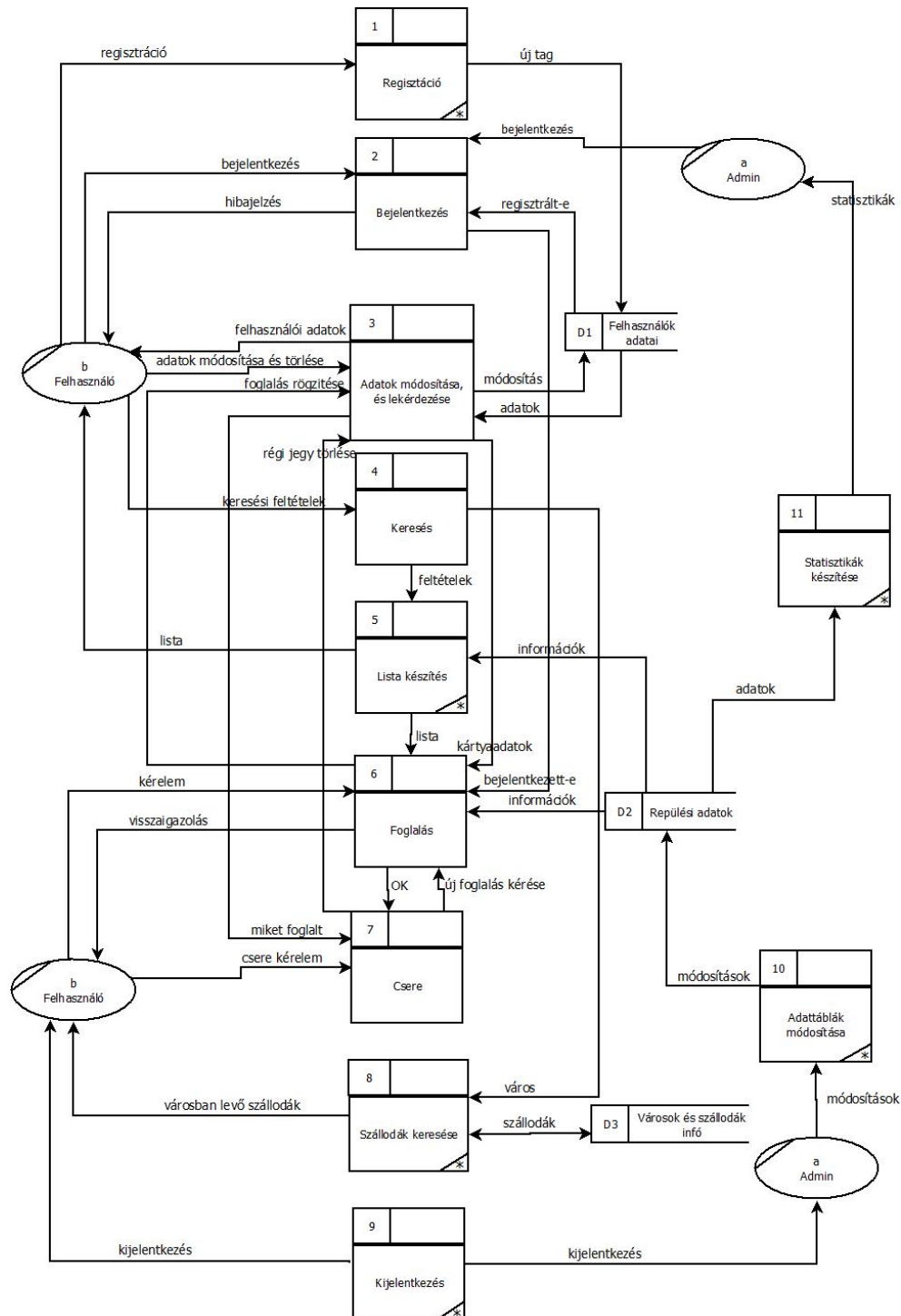
Nulladik szint:



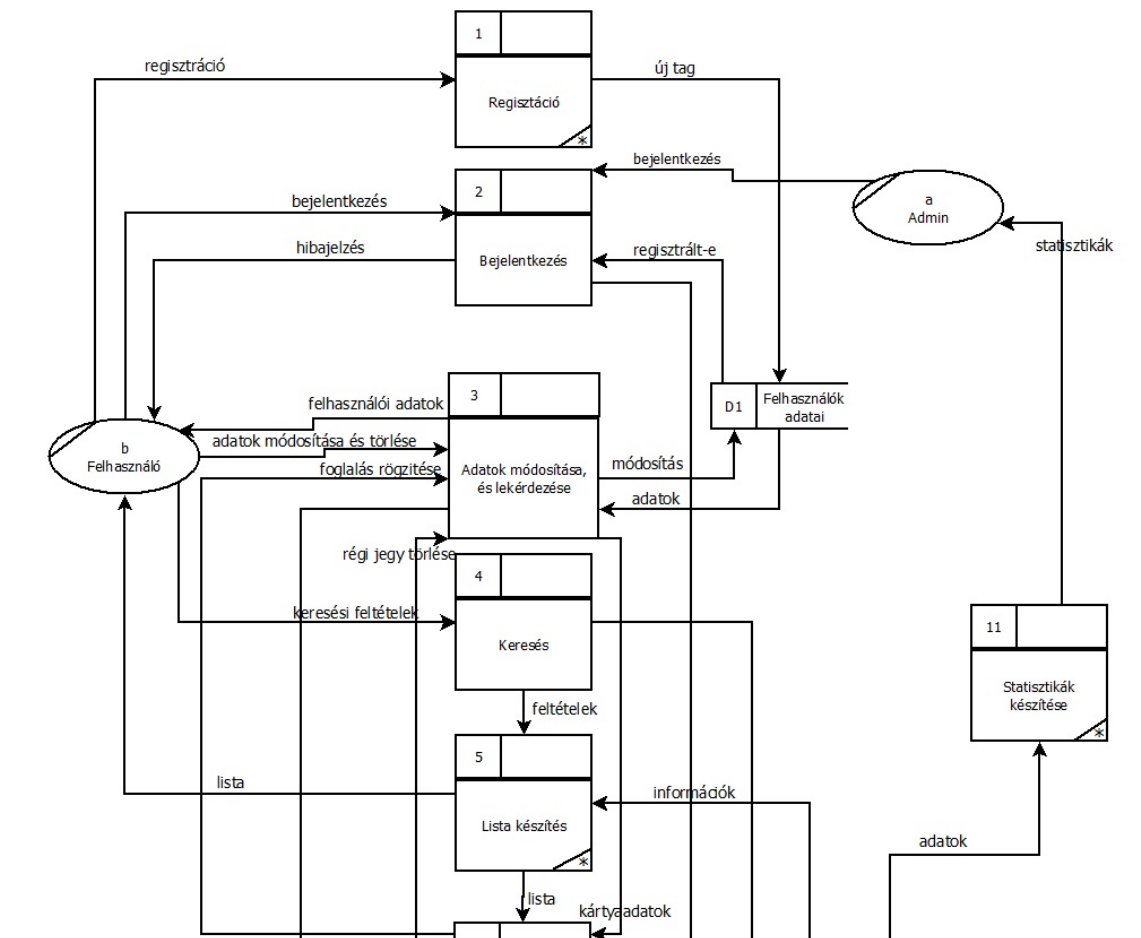
Első szint:

A jobb áttekinthetőség érdekében készítettem több képet az első szintről.

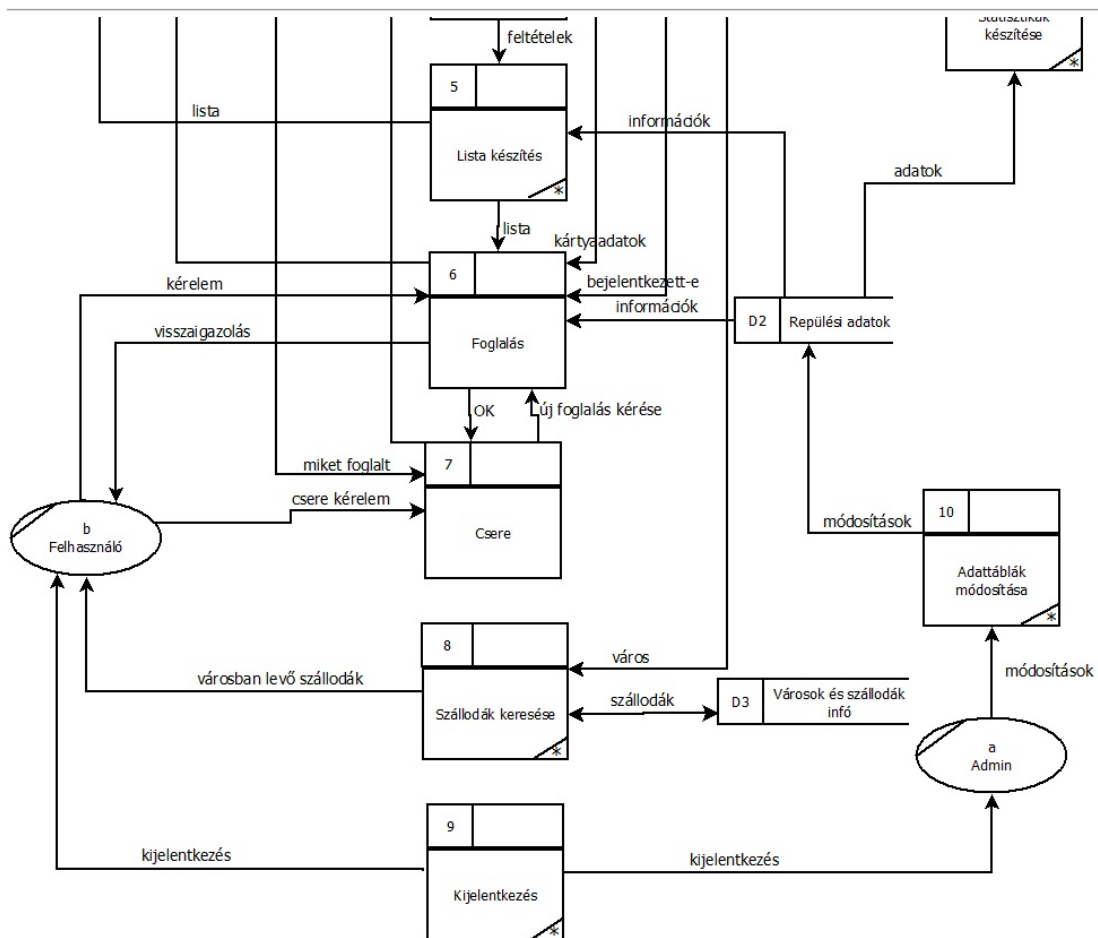
- Teljes első szint:



- Felső része nagytva:

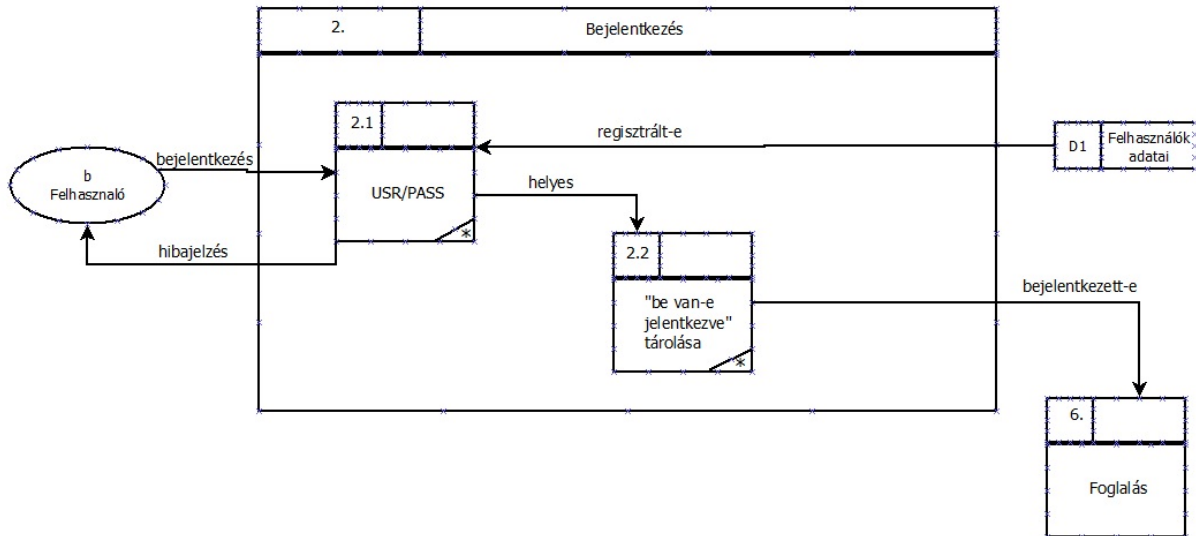


- Alsó része (nagyítva)

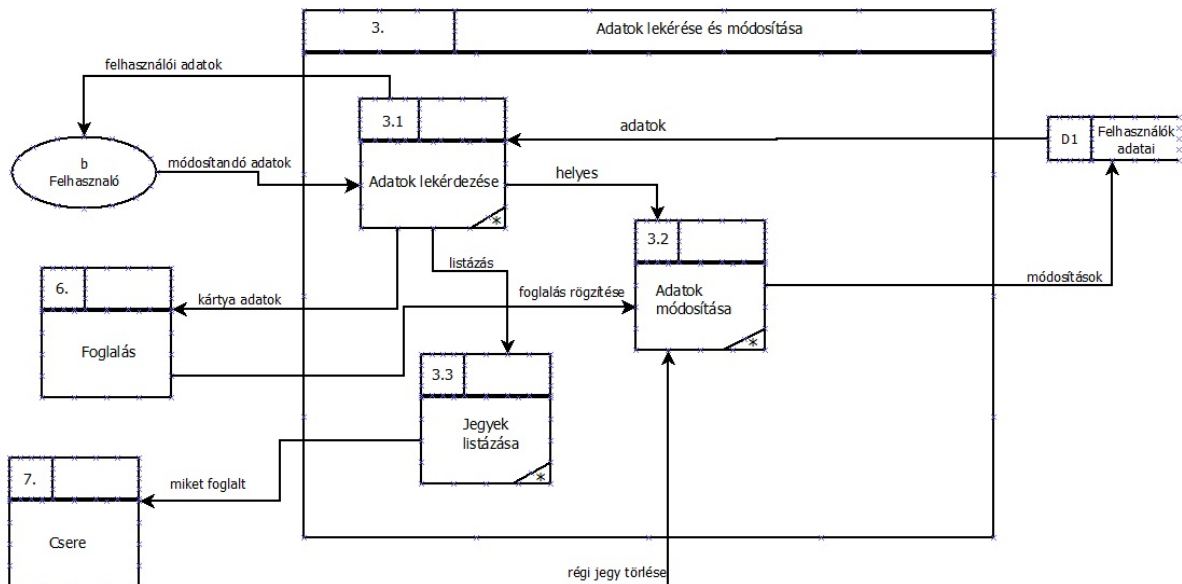


Második szint:

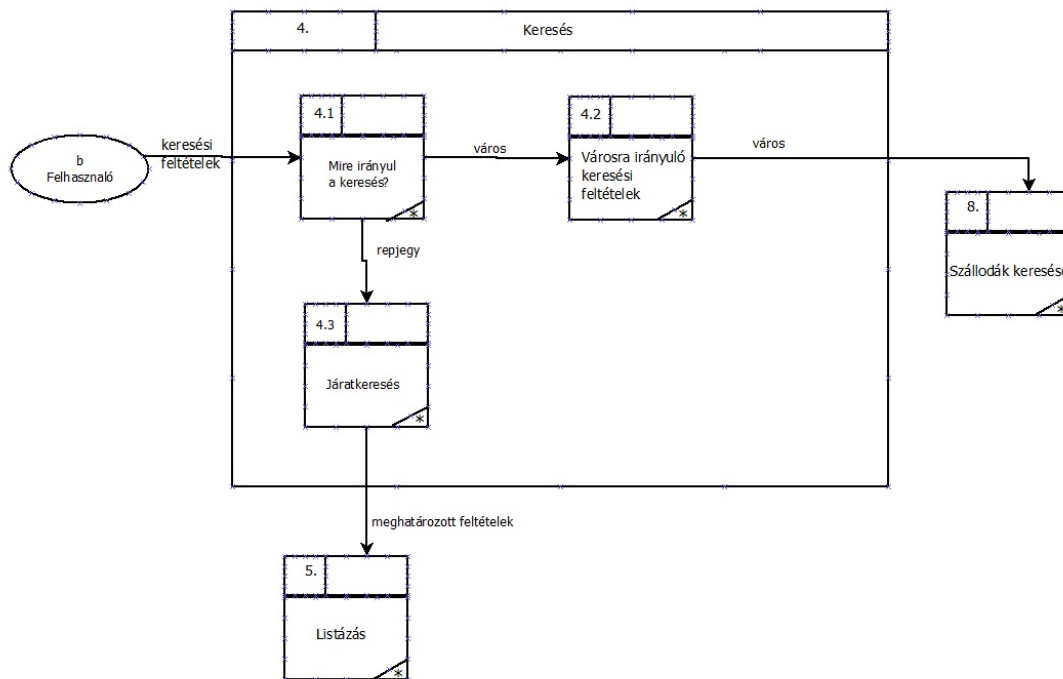
- Második folyamat kifejtése:



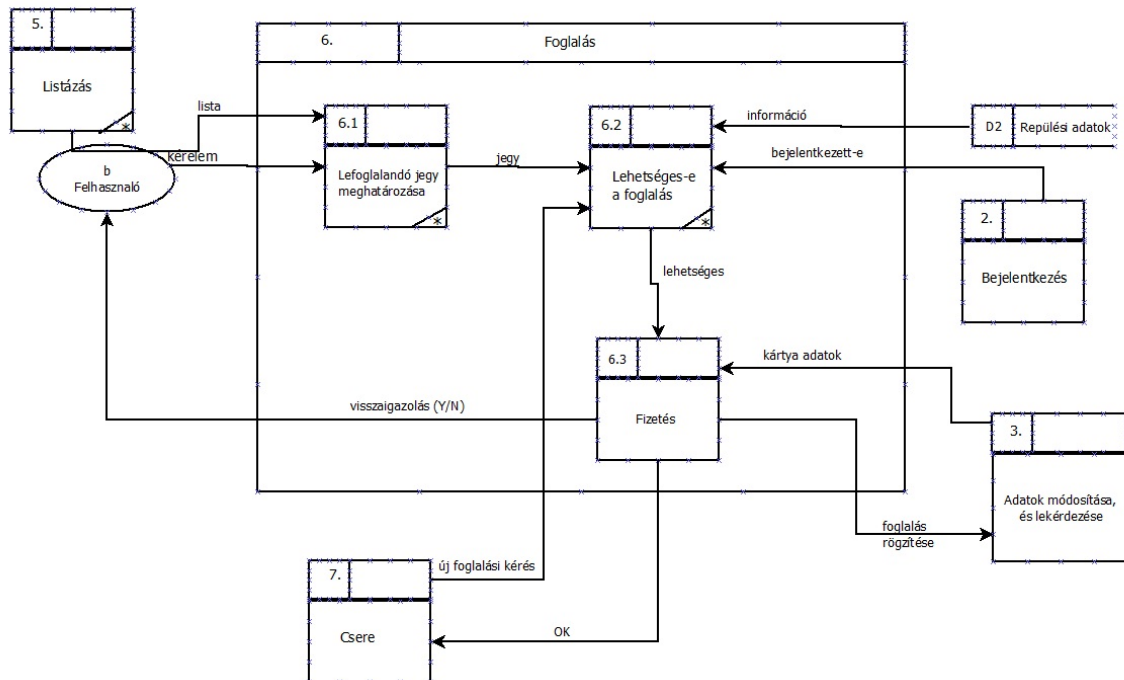
- Harmadik folyamat kifejtése:



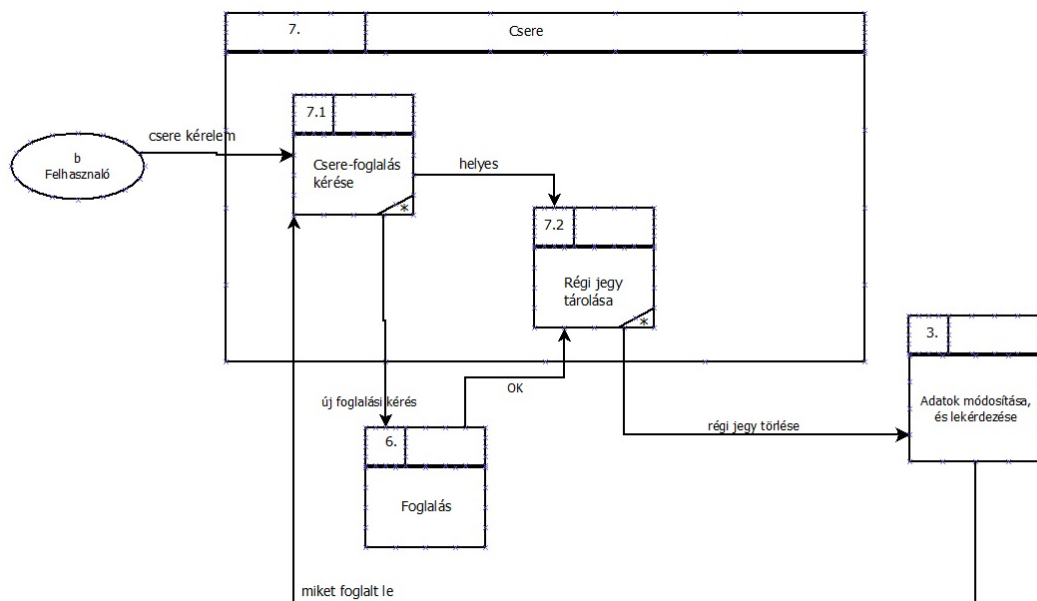
- Negyedik folyamat kifejtése:



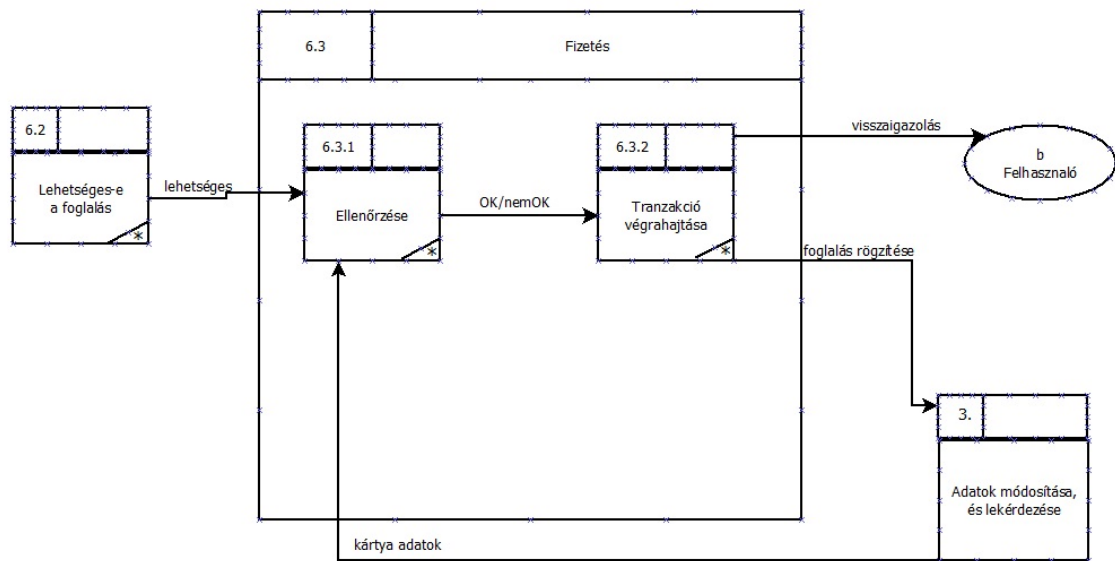
- Hatodik folyamat kifejtése:



- Hetedik folyamat kifejtése:



Harmadik szint:



Egyed-esemény mátrix

egyed	esemény	
felhasználó	L	regisztráció
foglalás	O	bejelentkezés
jegy	OMLT	adatok módosítása és lekérdezése
járat		keresés
menetrend		lista készítés
repülő		foglalás
repülési társaság		csere
nemzet		szállodák keresése
város		kijelentkezés
akciók		adattáblák módosítása
osztály		statisztikák készítése
biztosítás		
szálloda		

Szerep-funkció mátrix

admin	regisztrált	felhasználó	szerep / funkció
		X	regisztráció
X	X		bejelentkezés
	X		adatok módosítása és lekérdezése
	X	X	keresés
			lista készítés
	X		foglalás
	X		csere
	X	X	szállodák keresése
X	X		kijelentkezés
X			adattáblák módosítása
X			statisztikák készítése