

# Repülőjegy-foglalás, utazási iroda

Csapattagok:

- Bús Dávid
- Komlódi Tamás
- Tóth Péter Attila

Kurzus adatai:

Név: Adatbázis alapú rendszerek gyak.

Kód: IB152L-6

Időpont: Hétfő 12-14

Szemeszter: 2024-2025/2

## Specifikáció, részletes feladateleírás, követelménykatalógus

1. Rendszer célja A rendszer célja egy online repülőjegy-foglalási rendszer kifejlesztése, amely lehetővé teszi a felhasználók számára repülőjáratok keresését, foglalását és kapcsolódó szolgáltatások (pl. biztosítás, szállás) kezelését.

### 2. Funkcionális követelmények

#### 2.1. Repülőjáratok

- Repülőjáratok adatainak kezelése (pl. járatszám, útvonal, időpontok)

#### 2.2. Menetrend kezelése

- Járatok időpontjainak nyilvántartása

#### 2.3. Átszállások, csatlakozások kezelése

- Átszállások keresése
- Optimális csatlakozások keresése

#### 2.4. Repülőjárat-kereső

- Járatkereső dátum és csatlakozások figyelembevételével

#### 2.5. Repülőjegy-foglalás

- Járat kiválasztása
- Jegy típus választás
- Szabad helyek ellenőrzése
- Ülőhely kiválasztása

#### 2.6. Megvásárolt repülőjegyek adatai

- Felhasználó által megvásárolt jegyek listázása
- Jegy adatainak megtekintése

#### 2.7. Legnépszerűbb járatok

- Statisztikai adatok gyűjtése
- Legnépszerűbb úticél megjelenítése

#### 2.8. Kimutatások

- Havi/heti statisztikák készítése
- Utasszám és bevételek elemzése

#### 2.9. Jegykategóriák

- Jegytípusok kezelése (pl .Elő osztály... )
- Különböző jegykategóriák árainak meghatározása

#### 2.10. Repülőgép-típusok kezelése

- Repülőgép modellek adatainak nyilvántartása
- Kapacitások kezelése

#### 2.11. Biztosítók és biztosítási csomagok kezelése

- Biztosítási lehetőségek felkínálása foglalás során
- Különböző biztosítási csomagok kezelése

#### 2.12. Szállodák kezelése

- Célvárosban elérhető szállodák listázása

#### 2.13. Felhasználói fiók

- Regisztráció és bejelentkezés biztosítása
- Felhasználói adatok kezelése

#### 2.14. Adminisztrációs felület

- Járatok, biztosítási csomagok, szállodák adatainak feltöltése
- Statisztikai adatok kezelése

#### 2.15. Repülőgépek kezelése

- Repülőgépek kezelése (létrehozás, törlés, módosítás)

#### 2.16. Városok kezelése

- Városok kezelése (létrehozás, törlés, módosítás)

#### 2.17. Repülőgépmodellek kezelése

- Repülőgépmodellek kezelése (létrehozás, törlés, módosítás)

## Csapaton belül a munka felosztás

### 1. Mérföldkő

Bús Dávid

- Egyedmodell
- Egyed-kapcsolat diagram

- Egyed-esemény mátrix
- Funkciómeghatározás (funkcióleírás)

### Komlódi Tamás

- A csapaton belül a munka felosztásának részletes leírása szamon kérhető módon
- Egyed-kapcsolat diagram leképezése relációs adatbázissémákká
- Funkcionális függőségek felírása, relációsémák normalizálása 3NF-ig

### Tóth Péter Attila

- Címoldal
- Specifikáció, részletes feladatleírás, követelménykatalógus
- Logikai adatfolyam-diagramok
- Fizikai adatfolyam-diagramok
- Szerep-funkció mátrix

## 2. Mérföldkő

### Komlódi Tamás

- Adatbázist létrehozó szkript

## 3. Mérföldkő

### Bús Dávid

- Alapadatokat tartalmazó táblákhoz adatfelvitel, módosítás és törlés megvalósítása űrlapon keresztül
  - Biztosítások hozzáadása, lekérdezése, módosítása, törlése
  - Jegykategóriák hozzáadása, lekérdezése, módosítása, törlése
  - Repülők hozzáadása, lekérdezése, módosítása, törlése

### Komlódi Tamás

- Alapadatokat tartalmazó táblákhoz adatfelvitel, módosítás és törlés megvalósítása űrlapon keresztül
  - Szállodák hozzáadása, lekérdezése, módosítása, törlése
  - Jegyek-foglalások hozzáadása (2 tábla egyszerre van kezelve)

### Tóth Péter Attila

- Regisztrációs űrlap
- Bejelentkezési űrlap
- Alapadatokat tartalmazó táblákhoz adatfelvitel, módosítás és törlés megvalósítása űrlapon keresztül
  - Felhasználók hozzáadása, módosítása, törlése
  - Repülőgép modell hozzáadása, lekérdezése, módosítása, törlése
  - Járatok hozzáadása, lekérdezése, módosítása, törlése
  - Városok hozzáadása, lekérdezése, módosítása, törlése

## 4. Mérföldkő

### Bús Dávid

- Triggerek írása a specifikációban megjelölt funciókhoz:

- Jegyár limit
- Múltbéli járat tiltása
- Különböző városok járatok létrehozásánál
- Funkciókat megvalósító összetett lekérdezések:
  - Modell alapján átlagos ár
  - Járatok bevételi statisztikája
  - Utasok korosztálybeli megoszlása
  - Szolgáltatók bevétele
  - Dokumentációba megvalósítás helyének kigyűjtése

## Komlódi Tamás

- Alapadatokat tartalmazó táblák lekérdezése:
  - Járatok tábla
  - Foglalások tábla
  - Biztosítások tábla
  - Jegykategória tábla
  - Jegyek tábla
  - Modell tábla
  - Repülőgép tábla
  - Szállodák tábla
  - Város tábla
- Triggerek írása a specifikációban megjelölt funciókhoz:
  - Foglalt ülőhelyek tiltása
- Funkciókat megvalósító összetett lekérdezések:
  - Egyik városból a másikba menő járatok összeszámolása heti szinten
  - Egyik városból a másikba menő járatok összeszámolása havi szinten
  - Egyik városból a másikba menő emberek összeszámolása heti szinten
  - Egyik városból a másikba menő emberek összeszámolása havi szinten
  - Dokumentációba megvalósítás helyének kigyűjtése

## Tóth Péter Attila

- Alapadatokat tartalmazó táblák lekérdezése:
  - Felhasználók tábla
- Triggerek írása a specifikációban megjelölt funciókhoz:
  - Jövőbeli születésidátum tiltása
- Tárolt eljárások/függvények a megjelölt funciókhoz:
  - Jegyekszáma emberenként
  - Keresés
  - Legnépszerűbb járatok

## Végső Bemutatás:

Bús Dávid:

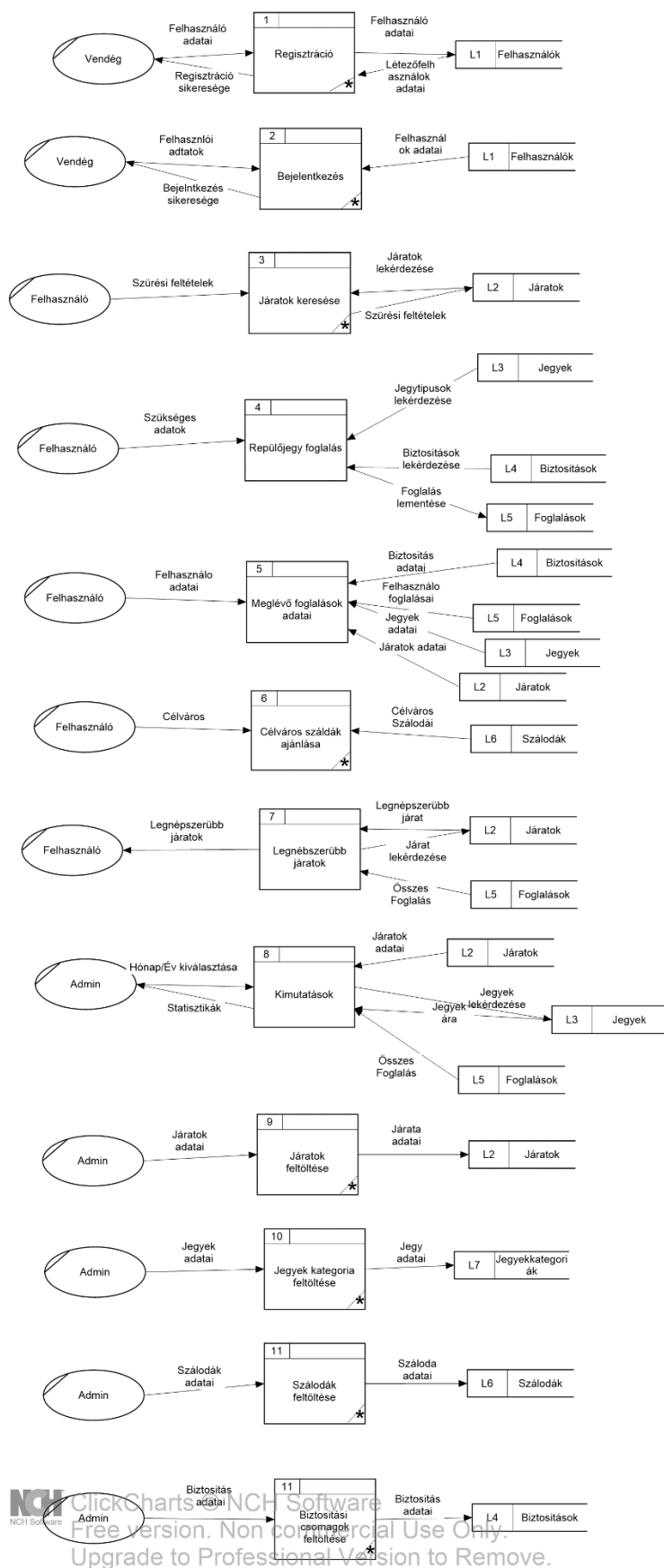
- Összetett lekérdezés:
  - Megjeleníti, hogy a foglalásokhoz milyen jegykategóriákat választottak
  - Kifestázza, hogy az egyes járatokon hány szabad hely van még

Komlódi Tamás:

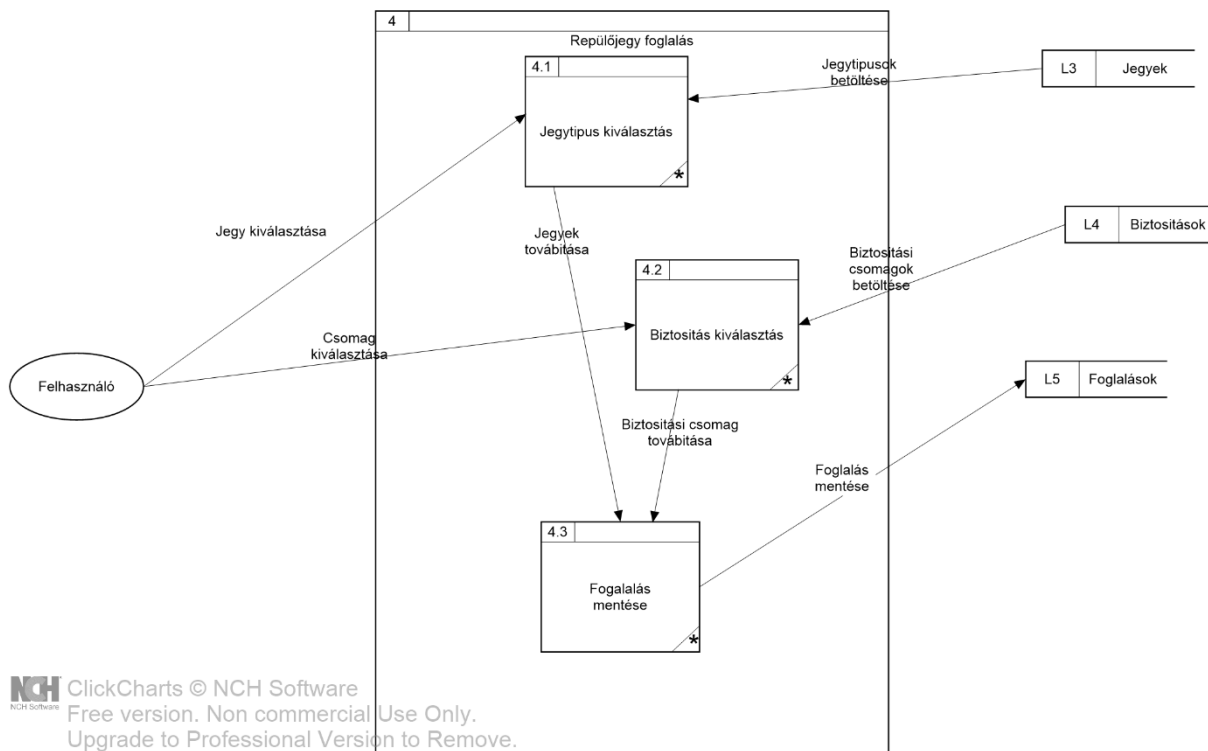
Tóth Péter Attila:

- Összetett lekérdezés:
  - Napi bevétel és eladott jegyszám
  - Az átlagnál drágább járatok

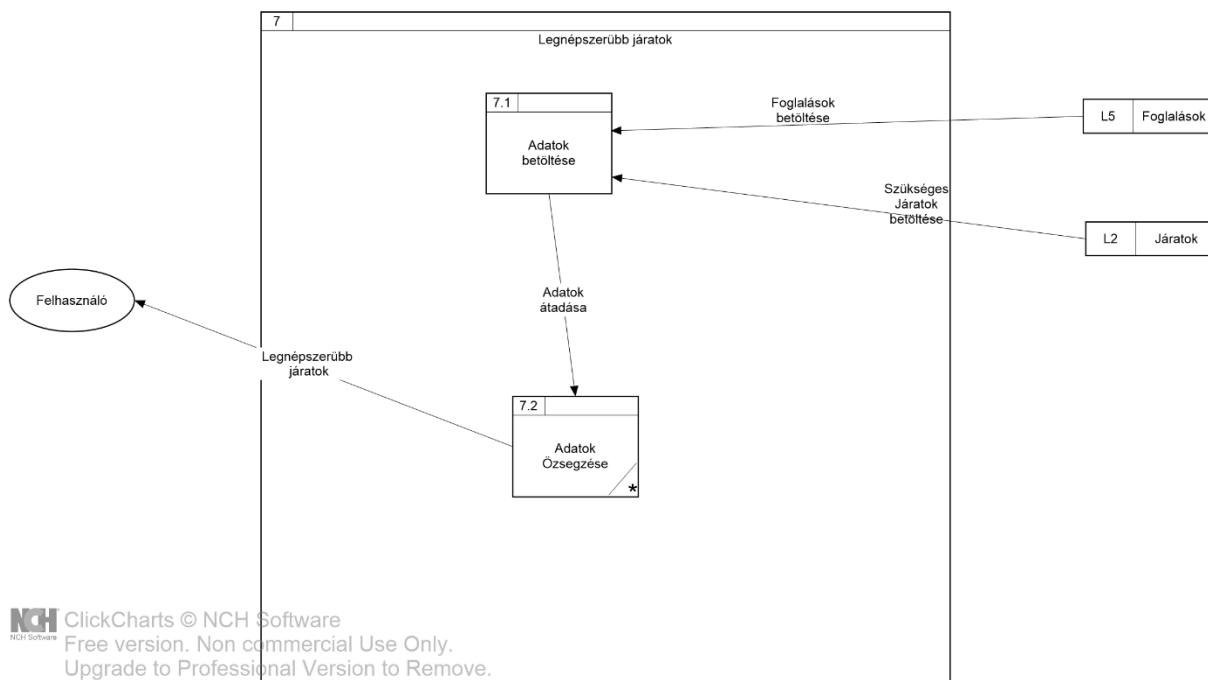
# Logikai adatfolyam-diagramok



Első szint

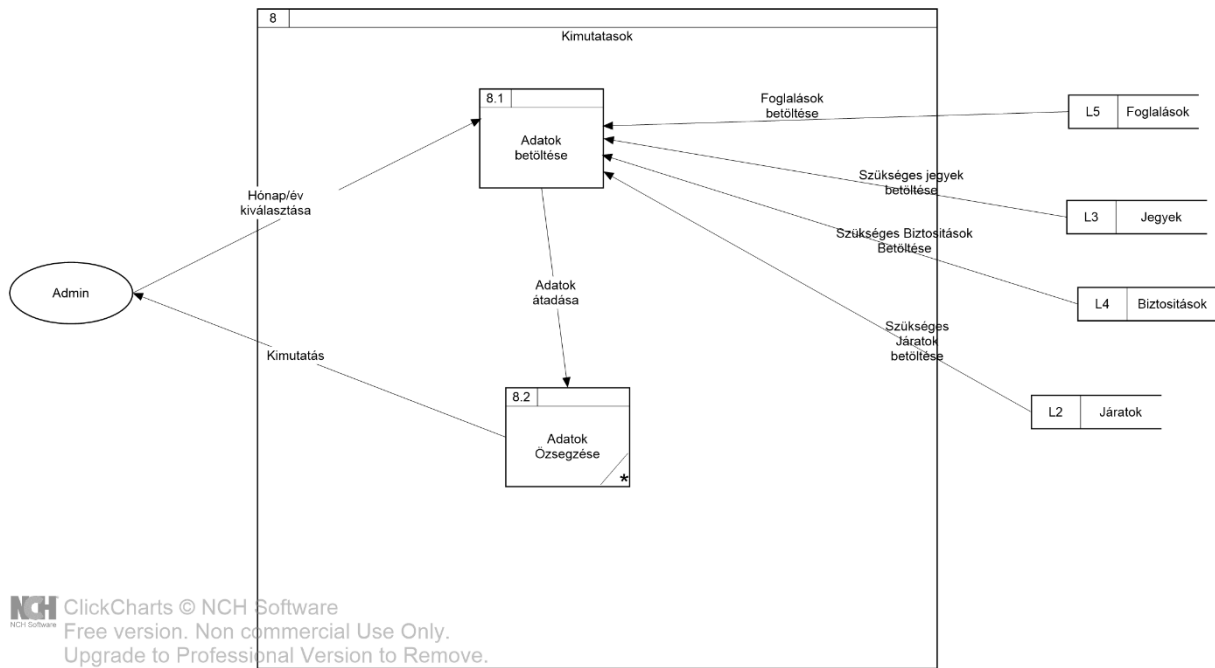


## 2. szint repülőjegy foglalás

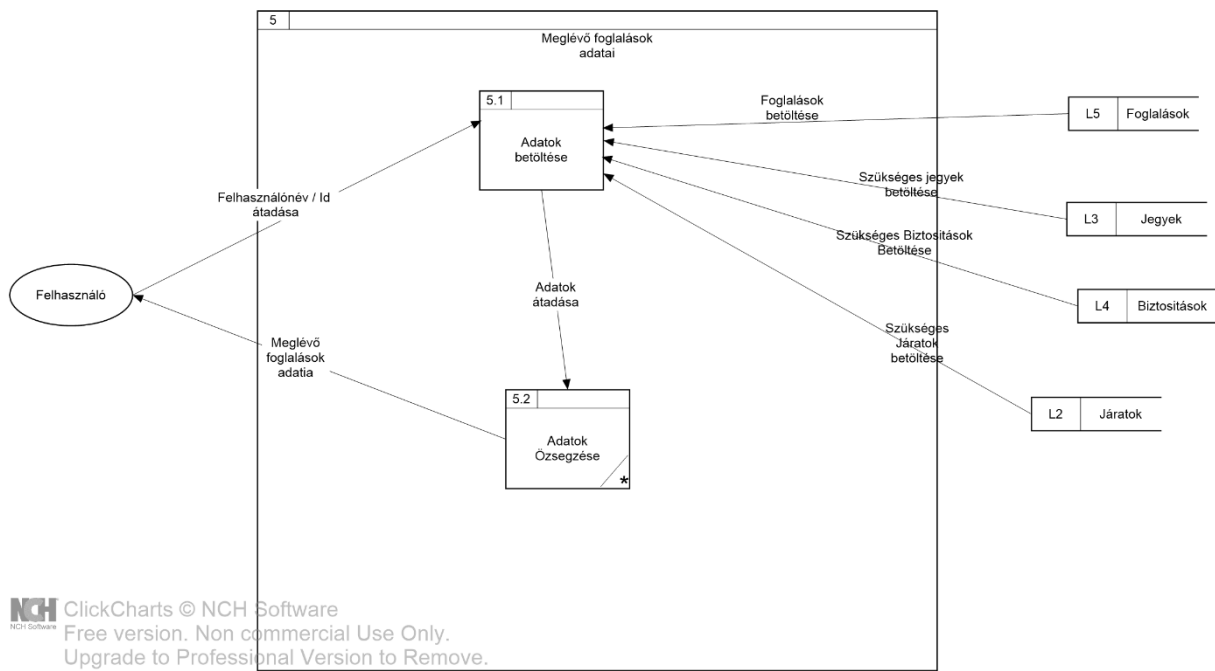


## 2. szint Legnépszerűbb járatok

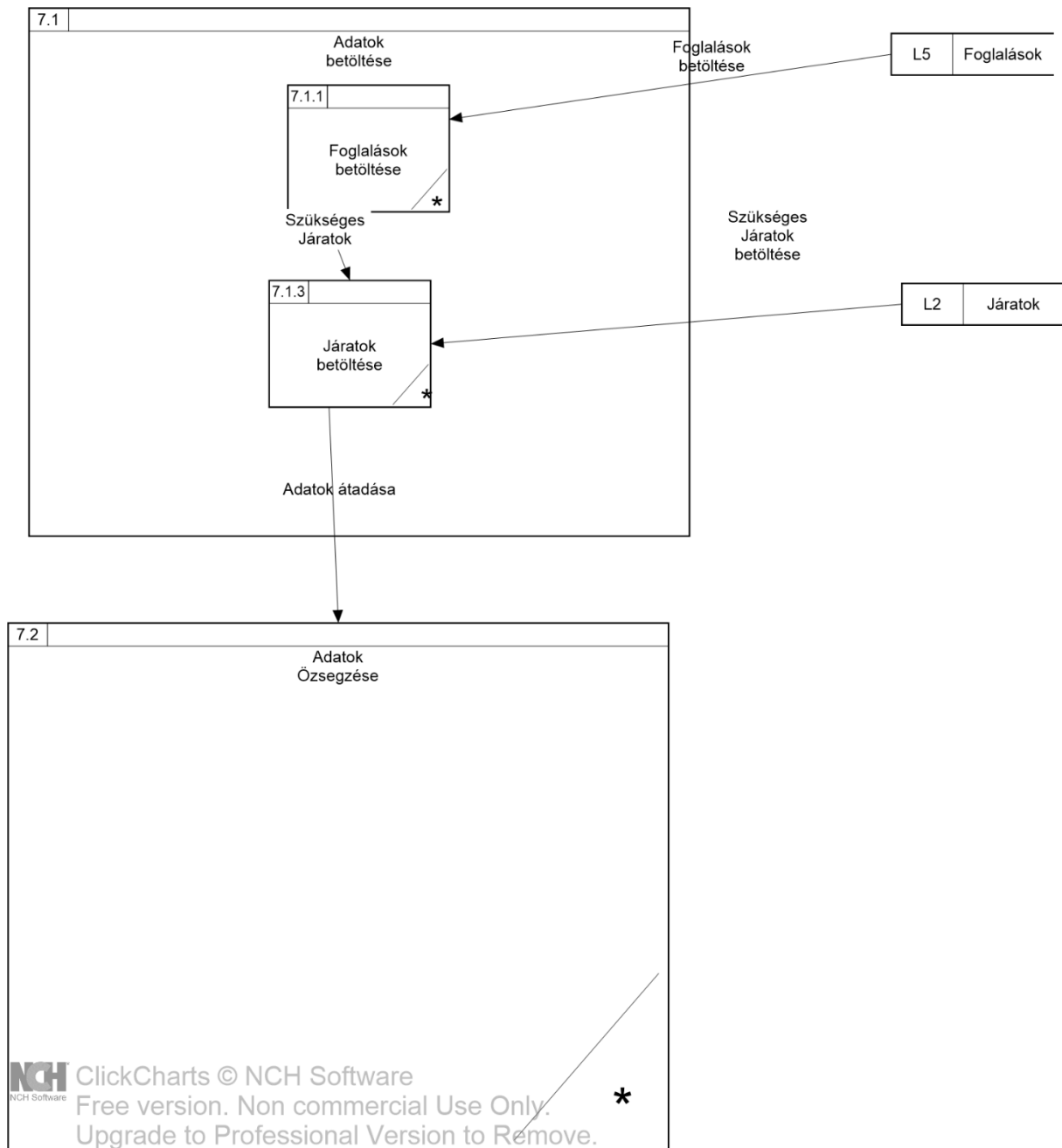




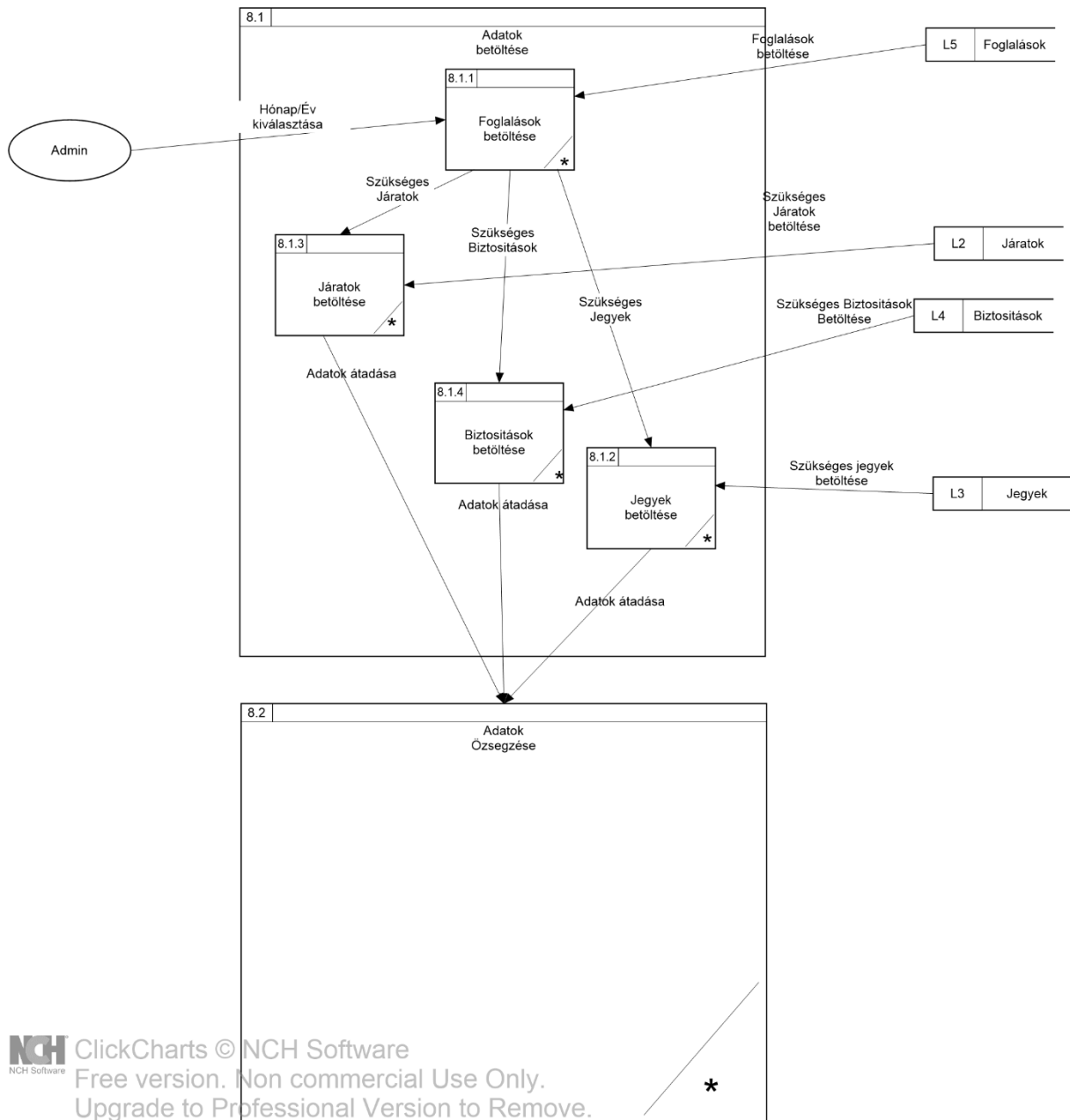
## 2. szint Kimutatások



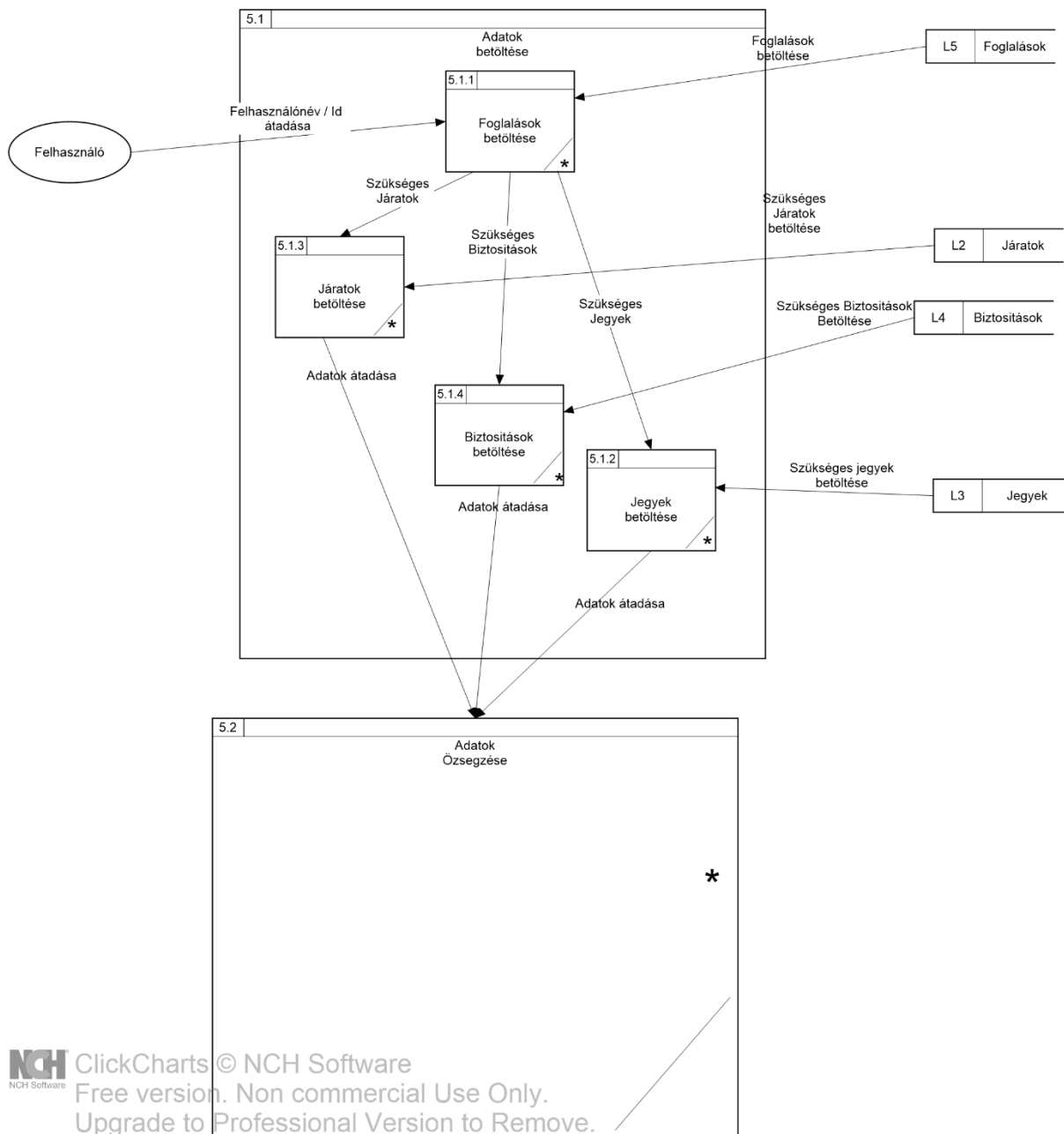
## 2. szint Meglévő foglalások



3. szint Legnépszerűbb adatok betöltése

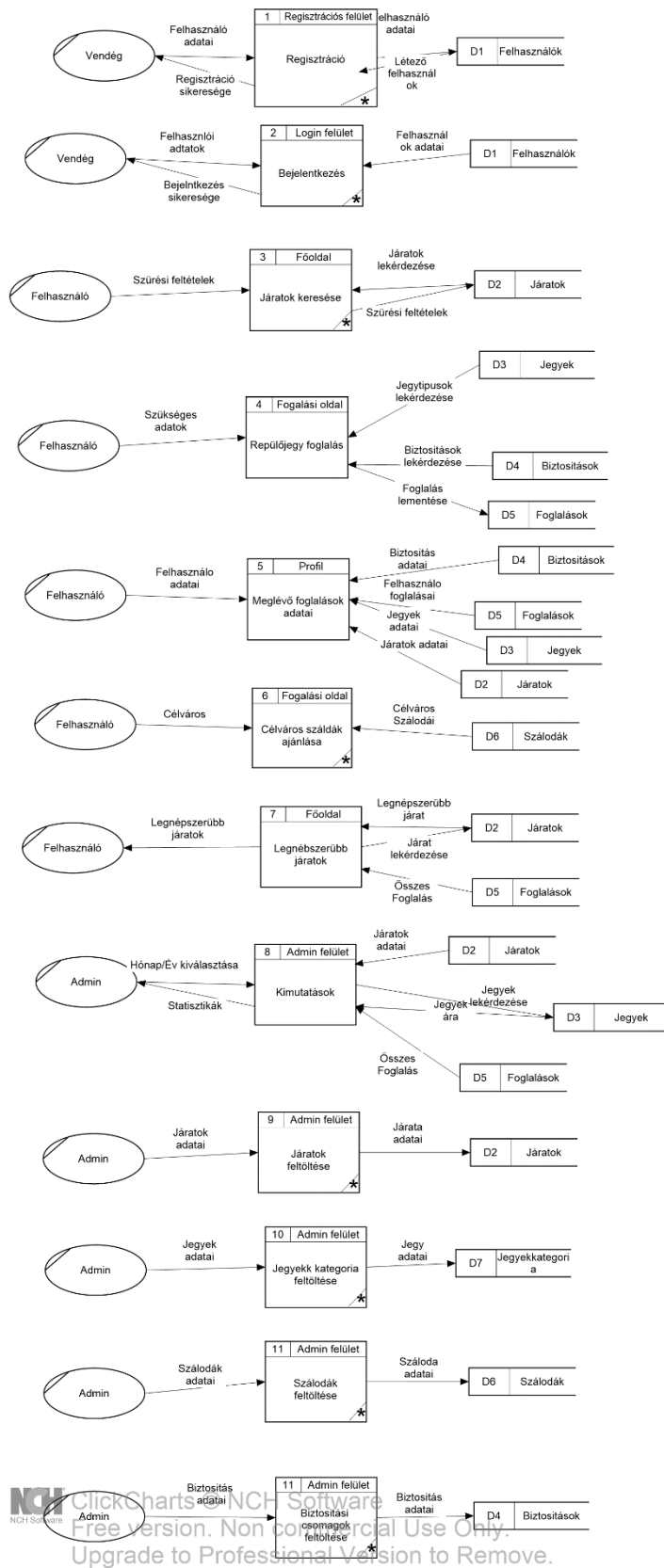


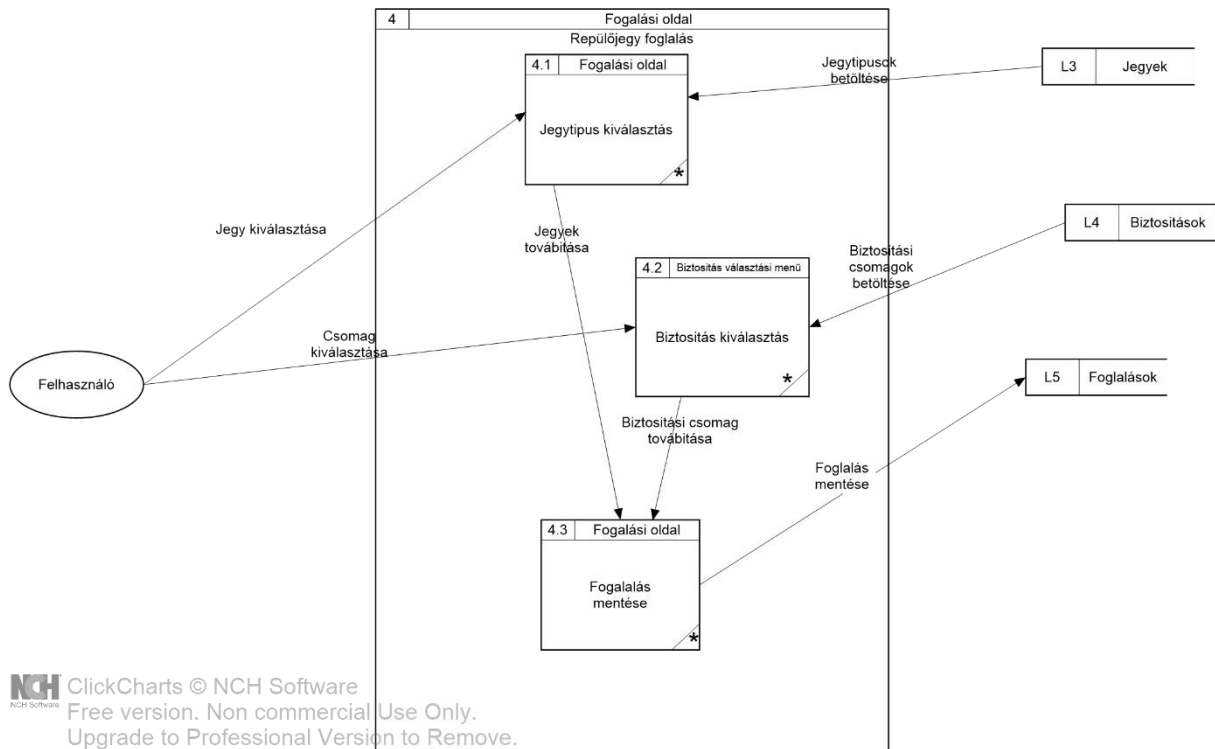
3. szint kimutatás adatok betöltése



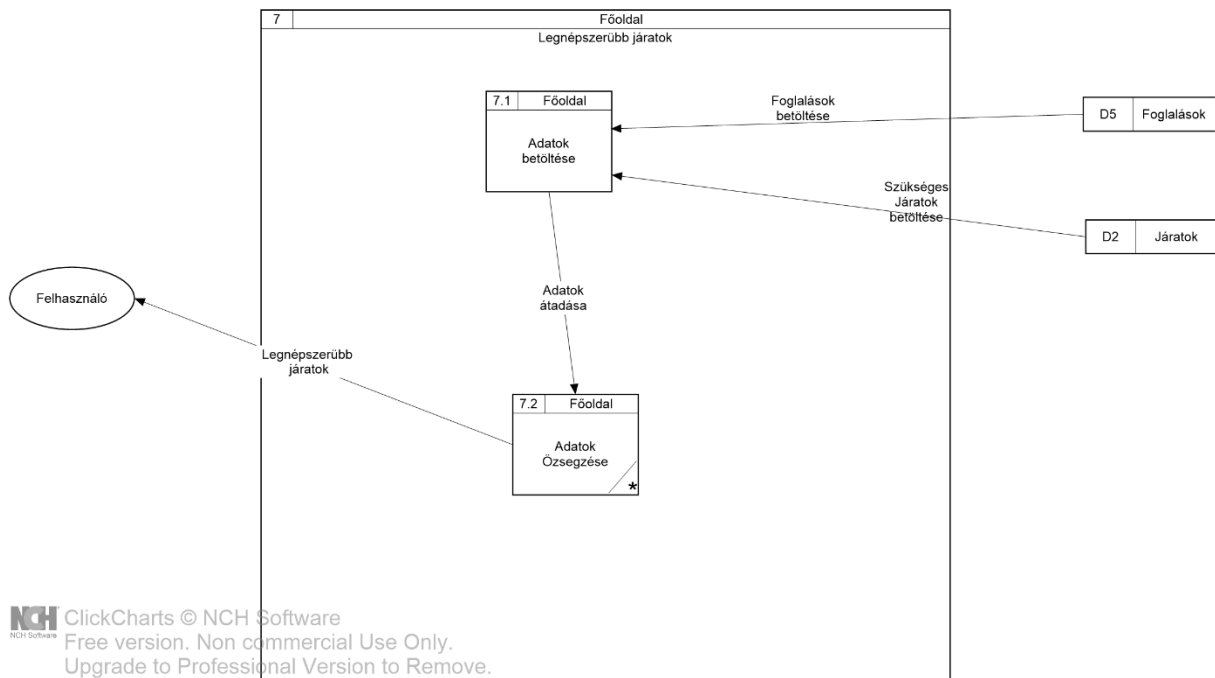
3. szint Adatok betöltése

# Fizikai adatfolyam-diagramok

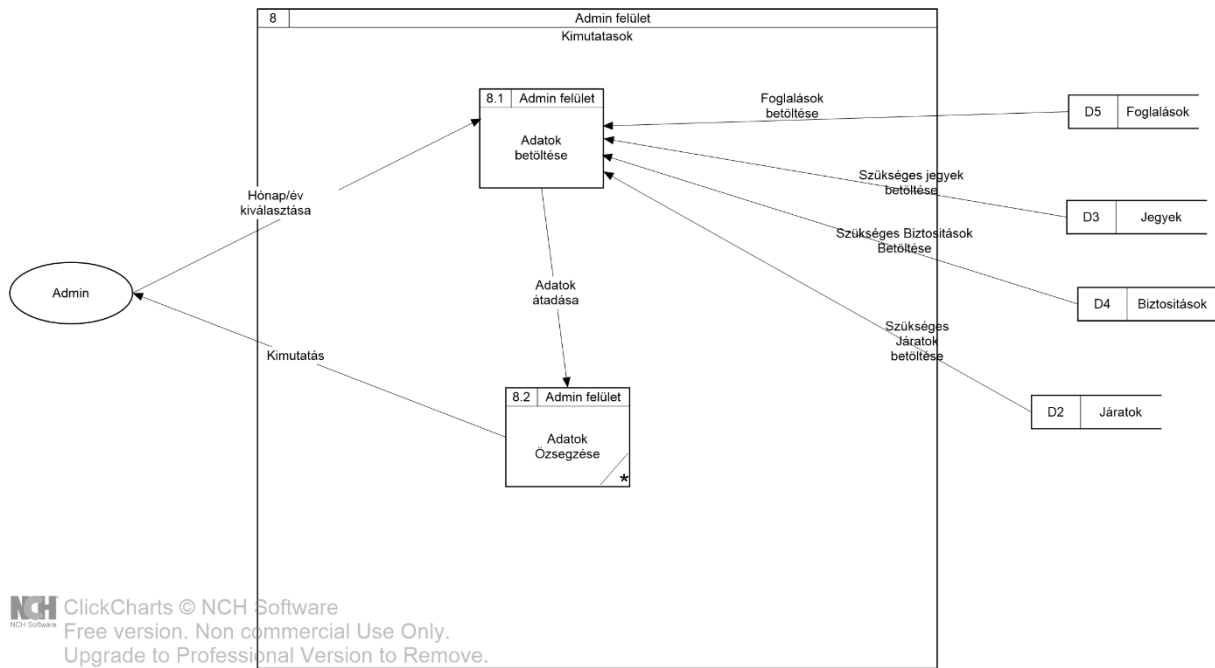




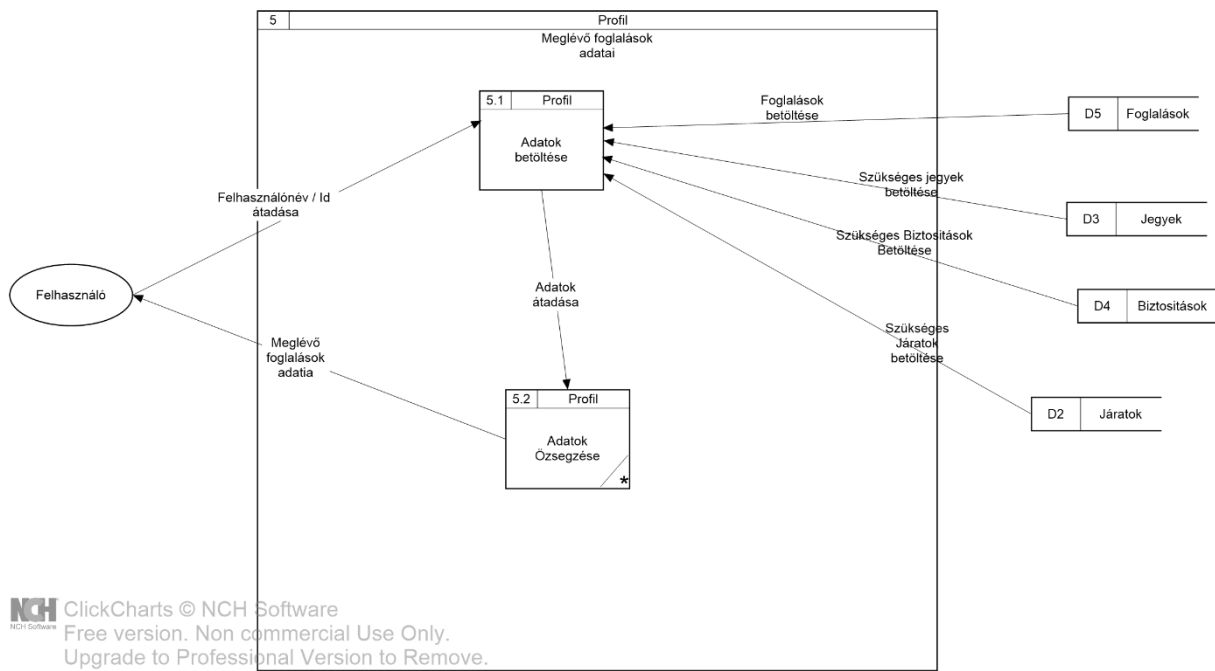
## 2. szint Repülőjegy foglalás



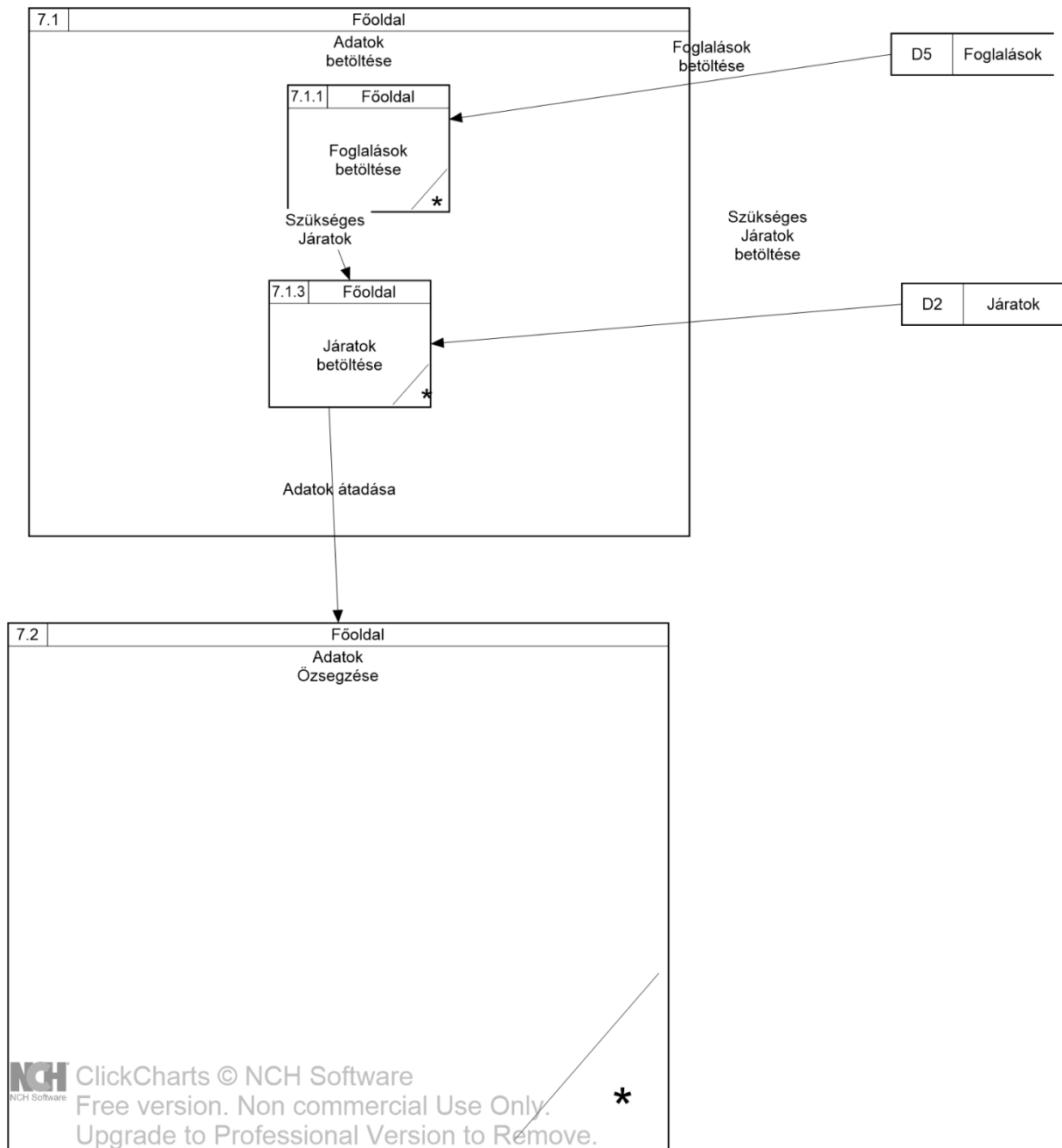
## 2. szint Legnépszerűbb járatok



## 2. szint Kimutatások

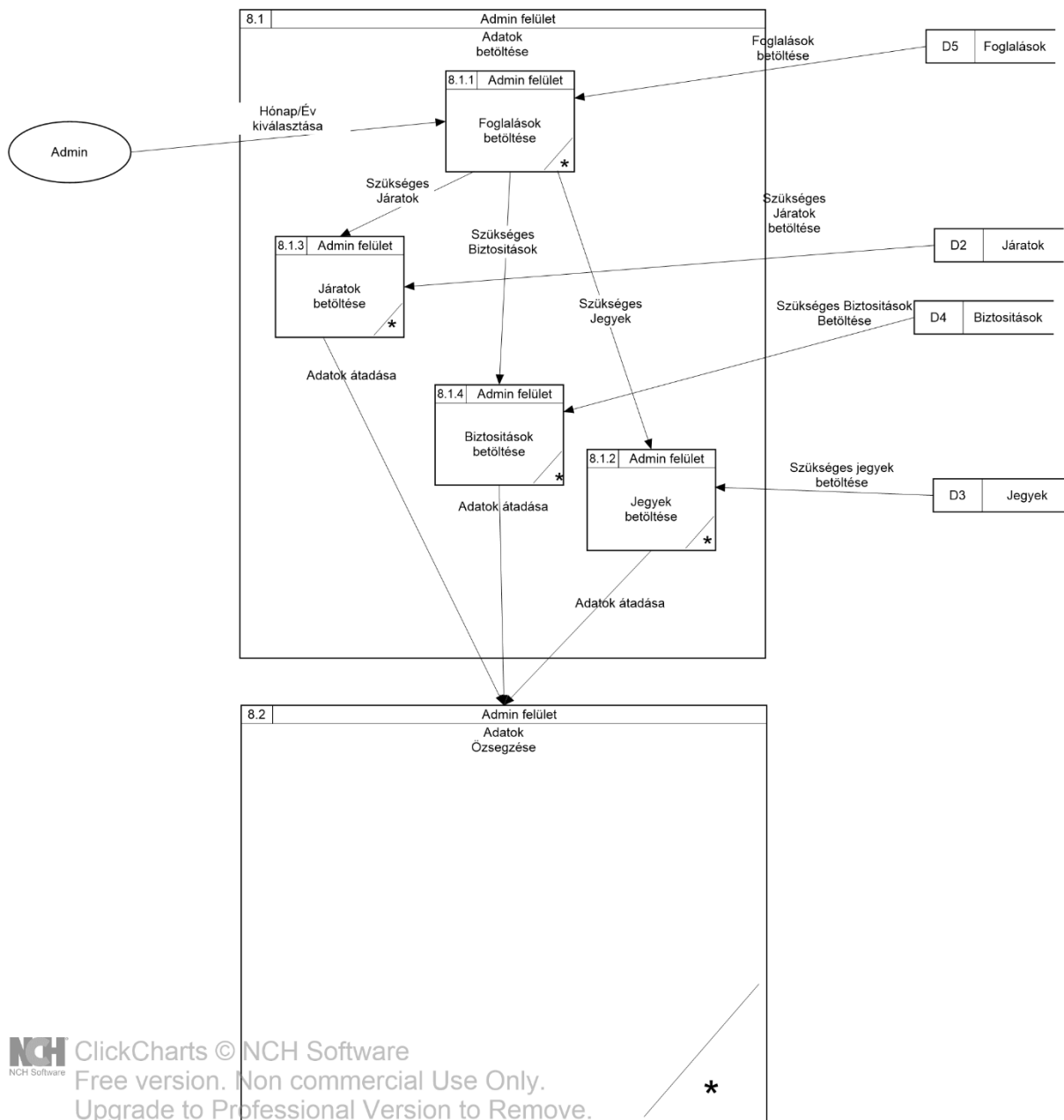


## 2. szint Foglalások



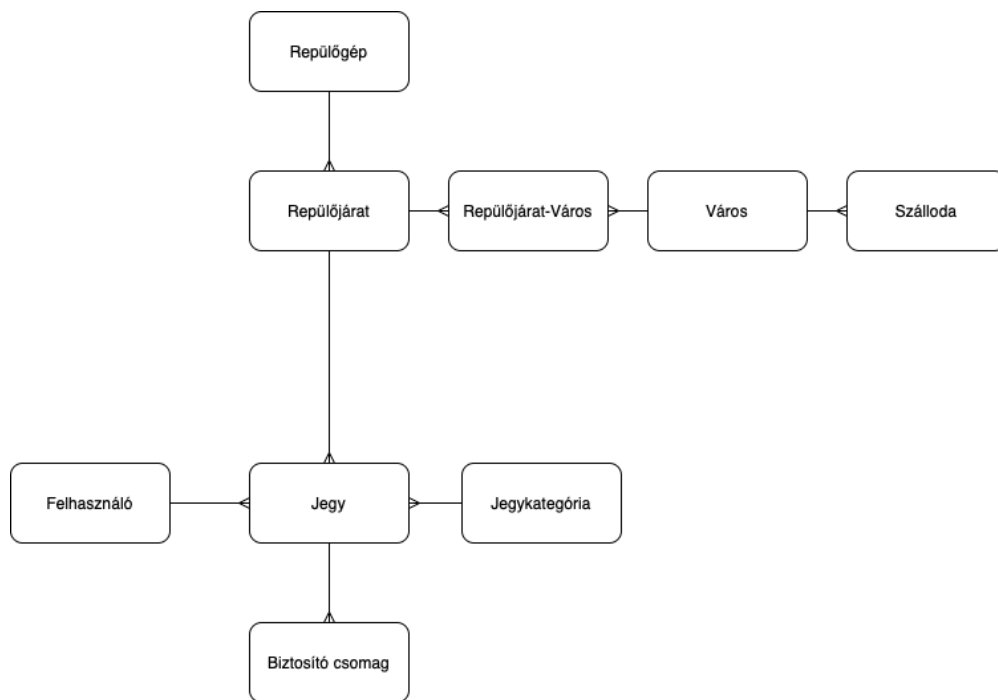
3. szint Adatok



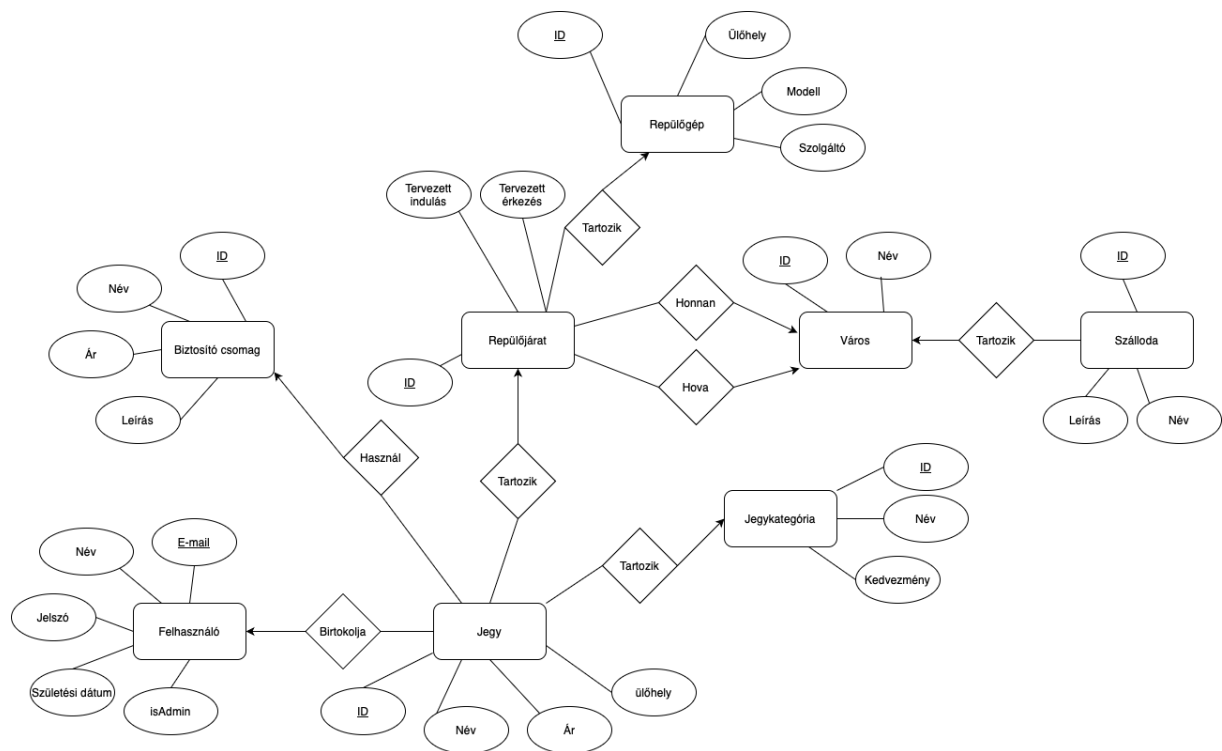




## Egyedmodell



## Egyed-kapcsolat diagram



Egyed-kapcsolat diagram leképezése relációs adatbázissémákra -

Funkcionális függőségek felírása, relációsémák normalizálása 3NF-ig

## Felülről lefelé történő elemzés (E-K diagramtól)

FELHASZNÁLÓK (email, név, jelszó, születési dátum, admin)

JÁRATOK (id, *repülő id*, *kiindulási hely*, *érkezési hely*, kiindulási időpont, érkezi időpont, ár)

- **repülő id** a **repülőgép** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE SET RESTRICT
- **kiindulási hely** a **város** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE RESTRICT
- **érkezési hely** a **város** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE RESTRICT

REPÜLŐGÉP (id, ülőhelyek száma, szolgáltató, modell, név)

JEGYEK (id, *járat id*, *biztosítás id*, *jegykategória id*, ülőhely, név, *email*)

- **járat id** a **járatok** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE SET NULL
- **biztosítás id** a **biztosítások** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE SET NULL
- **jegykategória id** a **jegykategóriák** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE SET NULL
- **email** a **felhasználók** egyedben az **email**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE CASCADE

JEGYKATEGÓRIA (id, név, kedvezmény)

BIZTOSÍTÁSOK (id, név, ár, leírás)

SZÁLLODÁK (id, *város id*, név, leírás)

- **város id** a **város** egyedben az **id**-re külső kulcs
  - ON UPDATE CASCADE
  - ON DELETE SET CASCADE

VÁROS (id, név)

# Alulról felfelé történő elemzés (funkcionális függőségek alapján)

## FELHASZNÁLÓK

- {email} -> {név, jelszó, születési dátum, admin}
- Feltételezzük, hogy lehet azonos névvel, születési dátummal, jelszóval és adminként felhasználók, viszont egyedi email címet engedünk csak meg.
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

## JÁRATOK

- {id} -> {kiindulási hely, kiindulási időpont, érkezési hely, érkezési időpont, repülő id, ár}
- Feltételezzük azt, hogy indulhat ugyan abból a városból és mehet ugyan abba a városba, ugyanabban az időpontokban repülő és ugyanazért az árért.
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

## REPÜLŐ

- {id} -> {ülőhelyek száma, szolgáltató, modell}
- {modell} -> {név, ülőhelyek száma} **ÚJ TÁBLA**
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

## MODELL (**ÚJ TÁBLA**)

- {modell} -> {név, ülőhelyek száma}
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

## FOGLALÁSOK

- {id} -> {járat id, biztosítás id, ülőhely, név, email}
- {járat id, ülőhely} -> {biztosítás id, név, email} **ÚJ TÁBLA**
- Feltételezzük azt, hogy több olyan jegy lehet, ami ugyan arra a járatra, biztosításra, névre és emailre szól.

- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

#### JEGYEK (ÚJ TÁBLA)

- {járat id, ülőhely} -> {biztosítás id, név, email}
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

#### VÁROS

- {id} -> {név}
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

#### BIZTOSÍTÁSOK

- {id} -> {név, ár, leírás}
- Feltételezzük azt, hogy azonos névvel árral és leírással lehetnek biztosítások.
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

#### SZÁLLODÁK

- {id} -> {város id, név, leírás}
- Feltételezzük azt, hogy lehet ugyanabban a városban több ugyanazon a néven futó, leírással rendelkező szálloda.
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

#### JEGYKATEGÓRIA

- {id} -> {név, kedvezmény}
- Az adattagok atomiak (1NF), minden másodlagos attribútum teljesen függ bármely kulcstól (2NF), és nem áll fent tranzitív függőség (3NF).

## Változtatott táblák

REPÜLŐGÉP (id, ülőhelyek száma, szolgáltató, *modell*, név)

- REPÜLŐGÉP (id, szolgáltató, *modell*)
  - **modell** a **modell** egyedben az **modell**-re külső kulcs

- ON UPDATE CASCADE
- ON DELETE SET RESTRICT
- MODELL (modell, név, ülőhelyek száma)

JEGYEK (id, járat id, biztosítás id, ülőhely, név, email)

- FOGLALÁSOK (id, jegykategória id, járat id, ülőhely)
  - **járat id** a **járatok** egyedben az **id**-re külső kulcs
    - ON UPDATE CASCADE
    - ON DELETE CASCADE
  - **jegykategória id** a **jegykategóriák** egyedben az **id**-re külső kulcs
    - ON UPDATE CASCADE
    - ON DELETE SET NULL
  - **járat id és ülőhely** a **jegyek** egyedben az **járat id-re és ülőhelyre** külső kulcs
    - ON UPDATE CASCADE
    - ON DELETE CASCADE
- JEGYEK (járat id, ülőhely, biztosítás id, név, email)
  - **biztosítás id** a **biztosítások** egyedben az **id**-re külső kulcs
    - ON UPDATE CASCADE
    - ON DELETE SET NULL
  - **járat id** a **járatok** egyedben az **id**-re külső kulcs
    - ON UPDATE CASCADE
    - ON DELETE CASCADE
  - **email** a **felhasználók** egyedben az **email**-re külső kulcs
    - ON UPDATE CASCADE
    - ON DELETE CASCADE



FELHASZNÁLÓK	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
email	TRUE			VARCHAR2(100)
név				VARCHAR2(100)
jelszó				VARCHAR2(100)
születési dátum				DATE
admin				NUMBER(1)

JÁRATOK	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
kiindulási hely		TRUE		VARCHAR2(100)
kiindulási időpont				DATETIME
érkezési hely		TRUE		VARCHAR2(100)
érkezési időpont				DATETIME
repülő id		TRUE		INTEGER
ár				INTEGER

REPÜLŐGÉP	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
szolgáltató				VARCHAR2(100)
modell		TRUE		VARCHAR2(100)

MODELL	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
modell	TRUE			VARCHAR2(100)
név				VARCHAR2(100)
ülőhelyek száma				INTEGER

FOGLALÁS	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
járat id		TRUE		INTEGER
jegykategória id		TRUE	TRUE	INTEGER
ülőhely		TRUE		INTEGER

JEGYEK	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
járat id	TRUE			INTEGER
ülőhely	TRUE			INTEGER
biztosítás id		TRUE	TRUE	INTEGER
név				VARCHAR2(100)
email		TRUE		VARCHAR2(100)

JEGYKATEGÓRI A	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER

név				VARCHAR2(100)
kedvezmény				INTEGER

BIZTOSÍTÁSOK	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
név				VARCHAR2(100)
ár				INTEGER
leírás				VARCHAR2(256)

SZÁLLODÁK	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
város id		TRUE		INTEGER
név				VARCHAR2(100)
leírás				VARCHAR2(256)

VÁROS	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
név				VARCHAR2(100)

## Egyed-esemény mátrix

<b>Egyed-esemény mátrix</b> L=létrehozás O=olvasás M=módosítás T=törlés	Bejelentkezés	Regisztráció	Járatok keresése	Repülőjegy foglalás	Meglévő foglalások adatai	Célváros szállodák ajánlása	Legnépszerűbb járatok	Kimutatók	Járatok kezelése	Jegykategóriák kezelése	Szállodák kezelése	Biztosítások kezelése	Repülőgépek kezelése	Városok kezelése	Repülőmodellek kezelése
Jegy				LT	O			O							
Repülőgép			O	O				O	O				LMT		
Biztosító csomag				O				O				LMT			
Felhasználó	O	L		O	O										
Repülőjárat			O	O			O	O	LMT						
Város			O	O		O		O	O					LMT	
Szálloda				O		O		O			LMT				
Jegykategória				O				O		LM T					
Modell				O				O							LMT

## Szerep-funkció mátrix

[illegible]

## Funkciómeghatározás (funkcióleírás)

<b>Funkciónév:</b> Regisztráció		<b>Funkció azonosító:</b> 1	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> Jogosult: Vendég			
<b>Funkció leírása:</b> A vendég annak érdekében, hogy teljes mértékben tudja használni az adott programot, vagy jegyet vásároljon, ahhoz regisztrálnia kell. Regisztráció során meg kell adnia különböző adatok, amelyek mentésre kerülnek.			
<b>Hibakezelés:</b> Ha nincs megadva valamelyik kötelező adat a regisztrációs felületen, akkor erről egy üzenet ablak jelenik meg. Ha a felhasználó már létezik, akkor erről szintén egy üzenet ablak jelzi, hogy az adott felhasználó már létezik.			
<b>AFD-eljárások: 1</b>			
<b>Események:</b> Minden alkalommal, mikor új felhasználó regisztrál		Esemény gyakoriság: Egyszeri esemény, 1 felhasználóra	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.13			
Tömegszerűség: Naponta akár többször			
Kapcsolódó funkciók: 2			
Lekérdezések: Az adatbázisból, hogy létezik-e már az adott e-mail		Lekérdezés gyakorisága: Ahány használt e-mailt ad meg a vendég	
Közös feldolgozás: Az adatbázisból, hogy létezik-e már az adott e-mail			
Dialogusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Felhasználó adatok feltöltése	Célérték: 2 mp	Tartomány: 3-5 mp	Megjegyzések:

<b>Funkciónév:</b> Bejelentkezés		<b>Funkció azonosító:</b> 2	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> vendég			
<b>Funkció leírása:</b> A vendég bejelentkeztetése.			
<b>Hibakezelés:</b> Ha valamely mezőt nem tölti ki a felhasználó, akkor egy hiba üzenet jelenik meg, amelyen az áll hogy „Hibás adatok!”. Ugyan ez az üzenet fog megjelenni, ha a felhasználó rosszul írja be e-mail vagy jelszavát.			
<b>AFD-eljárások:</b> 2			
<b>Események:</b> Minden alkalommal mikor be kíván jelentkezni a felhasználó		Esemény gyakoriság: Több alkalommal, akárhányszor használja a programot	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.13			
Tömegszerűség: Naponta többször			
Kapcsolódó funkciók: 1			
<b>Lekérdezések:</b> Az adatbázisból, hogy létezik-e az adott felhasználó		Lekérdezés gyakorisága: Minden bejelentkezési próbálnál	
Közös feldolgozás: Létezik-e az adott felhasználó			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Adatok lekérése	Célérték: 2 mp	Tartomány: 3-5 mp	Megjegyzések:

<b>Funkciónév:</b> Járatok keresése		<b>Funkció azonosító:</b> 3	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> vendég, admin, felhasználó			
<b>Funkció leírása:</b> Járatok keresése, az adott paraméterekkel. Miután beállítottuk, hogy hova, mikor és milyen módon szeretnénk utazni, válaszként egy listát kapunk az adott ideális járatokból.			
<b>Hibakezelés:</b> Néhány adott paraméteres adat előre kiválasztható, ezért ha a felhasználó saját kezével írna választ, egy hiba ablak jelezné, hogy „Kérjük válasszon az adott opciók közül!”			
<b>AFD-eljárások:</b> 3			
<b>Események:</b> Minden alkalommal mikor járat keresés történik		Esemény gyakoriság: Felhasználótól függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.4			
Tömegszerűség: Naponta többször			
Kapcsolódó funkciók:			
Lekérdezések: Az adott választott összes járat		Lekérdezés gyakorisága: Minden keresésnél	
Közös feldolgozás: Járatok listázása			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Járatok listázása	Célérték: 5 mp	Tartomány: 6-12 mp	Megjegyzések:

<b>Funkciónév:</b> Repülőjegy foglalása		<b>Funkció azonosító:</b> 4	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> admin, felhasználó			
<b>Funkció leírása:</b> A kiválasztott járaton lévő jegy kiválasztása. Miután megtalálta a felhasználó a tökéletes járatot, lefoglalja arra a jegyét, kiválassza hozzá az adott biztosítási csomagokat és ülőhelyét, majd elmentjük adatait.			
<b>Hibakezelés:</b>			
<b>AFD-eljárások:</b> 4			
<b>Események:</b> Minden jegy vásárlásnál		<b>Esemény gyakoriság:</b> Minden jegyvásárlásnál	
<b>I/O leírások:</b>			
<b>I/O szerkezetek:</b>			
<b>Követelménykatalógus hivatkozás:</b> 2.5			
<b>Tömegszerűség:</b> Naponta többször			
<b>Kapcsolódó funkciók:</b>			
<b>Lekérdezések:</b> Minden adatot lekérünk az adott jegyhez		<b>Lekérdezés gyakorisága:</b> Minden vásárlásnál	
<b>Közös feldolgozás:</b> Összes pontos adat lekérése az adott kiválasztott jegyhez			
<b>Dialógusnevek:</b>			
<b>Szolgáltatási szint követelményei:</b>			
<b>Leírása:</b> Jegy mentése	<b>Célérték:</b> 3 mp	<b>Tartomány:</b> 4-6 mp	<b>Megjegyzések:</b>

<b>Funkciónév:</b> Meglévő foglалások adatai		<b>Funkció azonosító:</b> 5	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> felhasználó, admin			
<b>Funkció leírása:</b> Megtudja tekinteni a felhasználó a saját jegyeit. Minden fontos információt megtalál, az összes jegyét, és azon belül minden jegy információját.			
<b>Hibakezelés:</b>			
<b>AFD-eljárások:</b> 5			
<b>Események:</b> Mikor a felhasználó megtekinti jegyeit		<b>Esemény gyakoriság:</b> Minden megtekintésnél	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.6			
Tömegszerűség: Naponta többször			
Kapcsolódó funkciók:			
<b>Lekérdezések:</b> Az adott jegyek információi		<b>Lekérdezés gyakorisága:</b> Minden megtekintésnél	
Közös feldolgozás:			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Jegyek lekérése	Célérték: 3 mp	Tartomány: 3-5 mp	Megjegyzések:



<b>Funkciónév:</b> Célváros szállodák ajánlása		<b>Funkció azonosító:</b> 6	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> vendég, felhasználó, admin			
<b>Funkció leírása:</b> Jegy foglaláskor a program felajánl szállodákat az adott célvárosban, amik közül lehet választani, hogy hol szállna meg.			
<b>Hibakezelés:</b>			
<b>AFD-eljárások:</b> 6			
<b>Események:</b> Ha a felhasználó jegyet foglal		Esemény gyakoriság: Felhasználótól függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.12			
Tömegszerűség: Felhasználótól függő			
Kapcsolódó funkciók: 4			
Lekérdezések: Szállodák lekérdezése		Lekérdezés gyakorisága: Felhasználótól függő	
Közös feldolgozás: Szállodák lekérdezése			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Szállodák lekérdezése	Célérték: 2 mp	Tartomány: 3-5 mp	Megjegyzések:

<b>Funkciónév:</b> Legnépszerűbb járatok		<b>Funkció azonosító:</b> 7	
<b>Típus:</b> online, felhasználói			
<b>Felhasználói szerepek:</b> vendég, felhasználó, admin			
<b>Funkció leírása:</b> Ki listázza a legnépszerűbb járatokat, az alapján, hogy melyik város felkapott, és melyik járatra jelentkeznek sokan.			
<b>Hibakezelés:</b>			
<b>AFD-eljárások:</b> 7			
<b>Események:</b> Fő oldalon történik, tehát mindig		<b>Esemény gyakoriság:</b> Oldal megnyitásakkor	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.7			
Tömegszerűség: Naponta többször			
Kapcsolódó funkciók:			
Lekérdezések: Járatok adatai, és jegyek adatai		<b>Lekérdezés gyakorisága:</b> Naponta többször	
Közös feldolgozás: Járatok és jegyek lekérdezése			
Dialogusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Népszerű járatok listázása	Célérték: 4 mp	Tartomány: 5-6 mp	Megjegyzések:

<b>Funkciónév:</b> Kimutatások		<b>Funkció azonosító:</b> 8	
<b>Típus:</b> online, admin, karbantartó			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Havi/éves kimutatási statisztikák az admin felületre. Bevételek kiszámítása. Gyakori járatok és egyéb hasznos információ.			
<b>Hibakezelés:</b> Helytelen szűrés feltételnél hiba üzenet dob, hogy „Nem létezik adott statisztika!”. Kevés adat esetében is ez az üzenet jelenik meg.			
<b>AFD-eljárások:</b> 8			
<b>Események:</b> Admin függő		Esemény gyakoriság: Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.8			
Tömegszerűség: kb. 8-10 naponta			
Kapcsolódó funkciók:			
Lekérdezések: Az adott szűrés feltételes adatok listázása, vagy kivetítése		Lekérdezés gyakorisága: Admin függő	
Közös feldolgozás: Adatok lekérése az adott helyeken			
Dialogusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Statisztika	Célérték: 5 mp	Tartomány: 6-8 mp	Megjegyzések:

<b>Funkciónév:</b> Járatok feltöltése		<b>Funkció azonosító:</b> 9	
<b>Típus:</b> online, admin, karbantartó			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Az admin feltudja tölteni az adatbázist új járatokkal. Ezeket tudja módosítani vagy törölni is.			
<b>Hibakezelés:</b> Mivel sok külső kulcs létezik ebben a táblában, ezért ha nem a létező városokat vagy repülőgépeket válasszuk ki, hanem mi írunk be, akkor „Érvénytelen adat!” hiba üzenettel fogjuk jelezni. Ha a tervezett indulás nagyobb, mint a tervezett érkezés, akkor „Helytelen időrend beosztás!” üzenettel fog jelezni. Ugyan ez az üzenet fog megjelenni, ha a tervezett érkezés kisebb, mint a tervezet indulás.			
<b>AFD-eljárások:</b> 9			
<b>Események:</b> Járatok kezelése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.1, 2.14			
<b>Tömegszerűség:</b> Hetente			
<b>Kapcsolódó funkciók:</b>			
<b>Lekérdezések:</b> A repülőgépekre és városokra		<b>Lekérdezés gyakorisága:</b> Járatok kezelésénél	
Közös feldolgozás: Repülőgép és város adatok			
<b>Dialógusnevek:</b>			
<b>Szolgáltatási szint követelményei:</b>			
<b>Leírása:</b> Járatok kezelése	<b>Célérték:</b> 4 mp	<b>Tartomány:</b> 5-7 mp	<b>Megjegyzések:</b>

<b>Funkciónév:</b> Jegykategóriák feltöltése		<b>Funkció azonosító:</b> 10	
<b>Típus:</b> online, admin, karbantartói			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Jegykategóriák kezelése, milyen kedvezmények, hányféle.			
<b>Hibakezelés:</b> Üres mező esetén hiba üzenet fogad minket, hogy „Kérem minden mezőt töltsön ki!”. Rossz adattípus esetén pedig „Nem megfelelő adat”			
<b>AFD-eljárások:</b> 10			
<b>Események:</b> Jegykategóriák feltöltése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.9			
Tömegszerűség: Havi 1x			
Kapcsolódó funkciók:			
<b>Lekérdezések:</b> Jegykategóriák adatai		<b>Lekérdezés gyakorisága:</b> Admin függő	
Közös feldolgozás:			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Jegyek feltöltése	Célérték: 2 mp	Tartomány: 3-5 mp	Megjegyzések:

<b>Funkciónév:</b> Szállodák feltöltése		<b>Funkció azonosító:</b> 11	
<b>Típus:</b> online, admin, karbantartói			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Az admin tudja kezelni a szállodákat. Hozzá tudni adni, módosítani vagy törölni.			
<b>Hibakezelés:</b> Hibás adatnál „Hibás adattípus!” üzenet fog megjelenni. Üres mezők esetén „Kérem töltsön ki minden mezőt!”.			
<b>AFD-eljárások:</b> 11			
<b>Események:</b> Szállodák kezelése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.14			
<b>Tömegszerűség:</b> Havonta 1x			
Kapcsolódó funkciók:			
<b>Lekérdezések:</b> Szállodák adatai		<b>Lekérdezés gyakorisága:</b> Minden kezelésnél	
Közös feldolgozás: Szállodák lekérdezése			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
<b>Leírása:</b> Szállodák feltöltése	<b>Célérték:</b> 4 mp	<b>Tartomány:</b> 6-8 mp	<b>Megjegyzések:</b>

<b>Funkciónév:</b> Biztosítási csomagok feltöltése		<b>Funkció azonosító:</b> 12	
<b>Típus:</b> online, admin, karbantartói			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Az admin tudja kezelni a biztosításokat. Hozzá tudni adni, módosítani vagy törölni.			
<b>Hibakezelés:</b> Hibás adatnál „Hibás adattípus!” üzenet fog megjelenni. Üres mezők esetén „Kérem töltsön ki minden mezőt!”.			
<b>AFD-eljárások:</b> 12			
<b>Események:</b> Biztosítás kezelése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.11-2.14			
<b>Tömegszerűség:</b> Havonta 1x			
Kapcsolódó funkciók:			
<b>Lekérdezések:</b> Biztosítás adatai		<b>Lekérdezés gyakorisága:</b> Minden kezelésnél	
Közös feldolgozás: Szállodák lekérdezése			
Dialógusnevek:			
<b>Szolgáltatási szint követelményei:</b>			
<b>Leírása:</b> Biztosítás feltöltése	<b>Célérték:</b> 4 mp	<b>Tartomány:</b> 6-8 mp	<b>Megjegyzések:</b>

<b>Funkciónév:</b> Repülők kezelése		<b>Funkció azonosító:</b> 13	
<b>Típus:</b> online, admin, karbantartói			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Az admin tudja kezelni a repülőket. Hozzá tudni adni, módosítani vagy törölni.			
<b>Hibakezelés:</b> Hibás adatnál „Hibás adattípus!” üzenet fog megjelenni. Üres mezők esetén „Kérem töltsön ki minden mezőt!”.			
<b>AFD-eljárások:</b> 11			
<b>Események:</b> Repülők kezelése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.15			
<b>Tömegszerűség:</b> Havonta 1x			
Kapcsolódó funkciók:			
<b>Lekérdezések:</b> Repülők adatai		<b>Lekérdezés gyakorisága:</b> Minden kezelésnél	
Közös feldolgozás: Repülők lekérdezése			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Repülők kezelése	Célérték: 4 mp	Tartomány: 6-8 mp	Megjegyzések:



<b>Funkciónév:</b> Városok kezelése		<b>Funkció azonosító:</b> 14	
<b>Típus:</b> online, admin, karbantartói			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Az admin tudja kezelni a városokat. Hozzá tudni adni, módosítani vagy törölni.			
<b>Hibakezelés:</b> Hibás adatnál „Hibás adattípus!” üzenet fog megjelenni. Üres mezők esetén „Kérem töltsön ki minden mezőt!”.			
<b>AFD-eljárások:</b> 11			
<b>Események:</b> Városok kezelése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
<b>Követelménykatalógus hivatkozás:</b> 2.16			
<b>Tömegszerűség:</b> Havonta 1x			
<b>Kapcsolódó funkciók:</b>			
<b>Lekérdezések:</b> Városok adatai		<b>Lekérdezés gyakorisága:</b> Minden kezelésnél	
<b>Közös feldolgozás:</b> Szállodák lekérdezése			
<b>Dialógusnevek:</b>			
<b>Szolgáltatási szint követelményei:</b>			
<b>Leírása:</b> Városok kezelése	<b>Célérték:</b> 4 mp	<b>Tartomány:</b> 6-8 mp	<b>Megjegyzések:</b>

<b>Funkciónév:</b> Repülőmodellek kezelése		<b>Funkció azonosító:</b> 14	
<b>Típus:</b> online, admin, karbantartói			
<b>Felhasználói szerepek:</b> admin			
<b>Funkció leírása:</b> Az admin tudja kezelni a modelleket. Hozzá tudni adni, módosítani vagy törölni.			
<b>Hibakezelés:</b> Hibás adatnál „Hibás adattípus!” üzenet fog megjelenni. Üres mezők esetén „Kérem töltsön ki minden mezőt!”.			
<b>AFD-eljárások:</b> 11			
<b>Események:</b> Repülőmodellek kezelése kezelése		<b>Esemény gyakoriság:</b> Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.17			
<b>Tömegszerűség:</b> Havonta 1x			
Kapcsolódó funkciók:			
<b>Lekérdezések:</b> Modellek adatai adatai		<b>Lekérdezés gyakorisága:</b> Minden kezelésnél	
Közös feldolgozás: Repülőmodellek lekérdezése			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
<b>Leírása:</b> Repülőmodellek kezelése	<b>Célérték:</b> 4 mp	<b>Tartomány:</b> 6-8 mp	<b>Megjegyzések:</b>

## Összetett lekérdezések

### Egyik városból a másikba menő járatok összeszámolása heti szinten

FlightDAO, 57. sor, Kimutatások funkció

```
SELECT
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'IW') AS TIME,
    BE.NEV AS STARTINGTOWN,
    KI.NEV AS LANDINGTOWN,
    COUNT(*) AS COUNT
FROM JARATOK
JOIN VAROS BE ON JARATOK.KIINDULASI_HELY = BE.ID
JOIN VAROS KI ON JARATOK.ERKEZESI_HELY = KI.ID
GROUP BY
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'IW'),
    BE.NEV,
    KI.NEV
ORDER BY TIME, STARTINGTOWN, LANDINGTOWN;
```

### Egyik városból a másikba menő járatok összeszámolása havi szinten

FlightDAO, 68. sor, Kimutatások funkció

```
SELECT
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'MM') AS TIME,
    BE.NEV AS STARTINGTOWN,
    KI.NEV AS LANDINGTOWN,
    COUNT(*) AS COUNT
FROM JARATOK
JOIN VAROS BE ON JARATOK.KIINDULASI_HELY = BE.ID
JOIN VAROS KI ON JARATOK.ERKEZESI_HELY = KI.ID
GROUP BY
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'MM'),
    BE.NEV,
    KI.NEV
ORDER BY TIME, STARTINGTOWN, LANDINGTOWN;
```

### Egyik városból a másikba menő emberek összeszámolása heti szinten

FlightDAO, 79. sor, Kimutatások funkció

```

SELECT
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'IW') AS TIME,
    BE.NEV AS STARTINGTOWN,
    KI.NEV AS LANDINGTOWN,
    COUNT(FOGLALAS.ID) AS COUNT
FROM JARATOK
JOIN VAROS BE ON JARATOK.KIINDULASI_HELY = BE.ID
JOIN VAROS KI ON JARATOK.ERKEZESI_HELY = KI.ID
JOIN FOGLALAS ON JARATOK.ID = FOGLALAS.JARAT_ID
GROUP BY
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'IW'),
    BE.NEV,
    KI.NEV
ORDER BY TIME, STARTINGTOWN, LANDINGTOWN;

```

### Egyik városból a másikba menő emberek összeszámolása havi szinten

FlightDAO, 91. sor, Kimutatások funkció

```

SELECT
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'MM') AS TIME,
    BE.NEV AS STARTINGTOWN,
    KI.NEV AS LANDINGTOWN,
    COUNT(FOGLALAS.ID) AS COUNT
FROM JARATOK
JOIN VAROS BE ON JARATOK.KIINDULASI_HELY = BE.ID
JOIN VAROS KI ON JARATOK.ERKEZESI_HELY = KI.ID
JOIN FOGLALAS ON JARATOK.ID = FOGLALAS.JARAT_ID
GROUP BY
    TRUNC(JARATOK.KIINDULASI_IDOPONT, 'MM'),
    BE.NEV,
    KI.NEV
ORDER BY TIME, STARTINGTOWN, LANDINGTOWN;

```

### Modell alapján átlagos ár

Plane\_ModelDAO, 29. sor, Kimutatások funkció

```

SELECT m.nev AS modell_nev, AVG(j.ar) AS atlag_ar
FROM jaratok j
LEFT JOIN repulogep r ON j.repulo_id = r.id
LEFT JOIN modell m ON r.modell = m.modell
GROUP BY m.nev
ORDER BY atlag_ar DESC

```

## Járatok bevételi statisztikája

TicketDAO, 73. sor, Kimutatások funkció

```
SELECT j.id AS jarat_id, v1.nev AS indulasi_varos, v2.nev AS celvaros,
j.kiindulasi_idopont, r.szolgaltato, COUNT(DISTINCT jegy.ulohely) AS eladott_jegyek, j.ar
AS alapar, SUM(NVL(b.ar, 0)) AS biztositas_bevetel, (COUNT(DISTINCT jegy.ulohely) *
j.ar) - SUM(j.ar * NVL(jk.kedvezmeny, 0) / 100) AS jegy_bevetel, (COUNT(DISTINCT
jegy.ulohely) * j.ar) - SUM(j.ar * NVL(jk.kedvezmeny, 0) / 100) + SUM(NVL(b.ar, 0)) AS
teljes_bevetel
FROM JARATOK j
JOIN VAROS v1 ON j.kiindulasi_hely = v1.id
JOIN VAROS v2 ON j.erkezesi_hely = v2.id
JOIN REPULOGEPI r ON j.repulo_id = r.id
LEFT JOIN JEGYEK jegy ON j.id = jegy.jarat_id
LEFT JOIN FOGLALAS f ON j.id = f.jarat_id AND jegy.ulohely = f.ulohely
LEFT JOIN JEGYKATEGORIA jk ON f.jegykatgoria_id = jk.id
LEFT JOIN BIZTOSITASOK b ON jegy.biztositas_id = b.id
GROUP BY j.id, v1.nev, v2.nev, j.kiindulasi_idopont, r.szolgaltato, j.ar
ORDER BY teljes_bevetel DESC"
```

## Utasok korosztálybeli megoszlása

UserDAO, 54. sor, Kimutatások funkció

```
SELECT
    CASE
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) < 18 THEN 'Kiskorú'
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) BETWEEN 18 AND 30 THEN 'Fiatal felnőtt'
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) BETWEEN 31 AND 50 THEN 'Középkorú'
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) BETWEEN 51 AND 65 THEN 'Idősebb felnőtt'
        ELSE 'Idős'
    END AS korcsoport,
COUNT(DISTINCT f.email) AS utasok_szama,
COUNT(DISTINCT j.jarat_id) AS repulesek_szama,
ROUND(AVG(j.ar), 2) AS atlagos_jegyar,
ROUND(AVG(COALESCE(b.ar, 0)), 2) AS atlagos_biztositas_ar
FROM FELHASZNALOK f
LEFT JOIN JEGYEK j ON f.email = j.email
LEFT JOIN JARATOK jar ON j.jarat_id = jar.id
```

```

LEFT JOIN BIZTOSITASOK b ON j.biztositas_id = b.id
GROUP BY
    CASE
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) < 18 THEN 'Kiskorú'
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) BETWEEN 18 AND 30 THEN 'Fiatal felnőtt'
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) BETWEEN 31 AND 50 THEN 'Középkorú'
        WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
f.szuletesi_datum) BETWEEN 51 AND 65 THEN 'Idősebb felnőtt'
        ELSE 'Idős'
    END
ORDER BY utasok_szama DESC

```

### Szolgáltatók bevétele

FlightDAO, 110. sor, Kimutatások funkció

```

SELECT r.szolgáltato, COUNT(DISTINCT jegy.ulohely || '-' || jegy.jarat_id) AS
osszes_eladott_jegy, SUM(j.ar - (j.ar * NVL(jk.kedvezmeny, 0) / 100)) AS jegy_bevetel
FROM REPULOGEP r
JOIN JARATOK j ON r.id = j.repulo_id
LEFT JOIN JEGYEK jegy ON j.id = jegy.jarat_id
LEFT JOIN FOGLALAS f ON jegy.ulohely = f.ulohely AND jegy.jarat_id = f.jarat_id
LEFT JOIN JEGYKATEGORIA jk ON f.jegykatgoria_id = jk.id
GROUP BY r.szolgáltato
ORDER BY jegy_bevetel DESC

```

### Napi bevétel és eladott jegyszám

FlightDAO 152.sor, Kimutatások funkció

```

SELECT
    TRUNC(j.kiindulasi_idopont) AS SZOLGALTATO,
    COUNT(f.id) AS OSSZES_ELADOTT_JEGY,
    SUM(j.ar) AS JEGY_BEVETEL
FROM
    FOGLALAS f
JOIN
    JARATOK j ON f.jarat_id = j.id
GROUP BY
    TRUNC(j.kiindulasi_idopont)
ORDER BY
    SZOLGALTATO

```

## Az átlagnál drágább járatok

PlaneDAO 36.sor , Kimutatások funkció

```
SELECT
    j.id,
    j.ar as modell
    r.szolgáltato
FROM
    JARATOK j
JOIN
    REPULOGEPI r ON j.repulo_id = r.id
WHERE
    j.ar > (
        SELECT AVG(j2.ar)
        FROM JARATOK j2
        WHERE j2.repulo_id = j.repulo_id
    )
ORDER BY
    j.ar DESC
```

## Megjeleníti, hogy a foglalásokhoz milyen jegykategóriákat választottak

TicketDAO 76. sor, getTicketStat (), Kimutatások

```
SELECT
    jk.nev AS jegykategoria,
    jk.kedvezmeny AS kedvezmeny_szazalek,
    COUNT(f.id) AS foglalasok_szama,
    ROUND(AVG(j.ar * (1 - jk.kedvezmeny/100)), 2) AS atlagos_fizetett_ar
FROM FOGLALAS f
JOIN JEGYKATEGORIA jk ON f.jegykategoria_id = jk.id
JOIN JARATOK j ON f.jarat_id = j.id
GROUP BY jk.nev, jk.kedvezmeny
ORDER BY foglalasok_szama DESC;
```

## Kilistázza, hogy az egyes járatokon hány szabad hely van még

TicketDAO 89. sor, getFreeSeats(), Kimutatások

```
SELECT
    j.id AS jarat_azonosito,
    vki.nev || ' -> ' || ver.nev AS utvonal,
    j.kiindulasi_idopont,
    m.ulohelyek_szama AS osszes_hely,
    COUNT(jegy.ulohely) AS foglalt_helyek,
```

```

        m.ulohelyek_szama - COUNT(jegy.ulohely) AS szabad_helyek
FROM JARATOK j
JOIN VAROS vki ON j.kiindulasi_hely = vki.id
JOIN VAROS ver ON j.erkezesi_hely = ver.id
JOIN REPULOGEP r ON j.repulo_id = r.id
JOIN MODELL m ON r.modell = m.modell
LEFT JOIN JEGYEK jegy ON j.id = jegy.jarat_id
GROUP BY j.id, vki.nev, ver.nev, j.kiindulasi_idopont, m.ulohelyek_szama
ORDER BY j.kiindulasi_idopont;

```

## Triggerek

### Jegyár limit

Adatbazis.sql, 793. sor, Járatok kezelése

```

CREATE OR REPLACE TRIGGER check_price_limit
BEFORE INSERT OR UPDATE ON JARATOK
FOR EACH ROW
BEGIN
    IF :NEW.ar > 100000 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Az ár túl magas!');
    END IF;
END;
/

```

### Foglalt ülőhelyek tiltása

Adatbazis.sql, 854. sor, Repülőjegy foglalás

```

CREATE OR REPLACE TRIGGER foglalt_ulohely
BEFORE INSERT
ON FOGLALAS
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM FOGLALAS F
    WHERE JARAT_ID = :NEW.JARAT_ID
    AND F.ULOHELY = :NEW.ULOHELY;
    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20005, 'Ez az ülőhely már foglalt erre a
        járatra!');
    END IF;

```



```
END;
```

```
/
```

### Jövőbeli születésidátum tiltása

Adatbazis.sql, 888. sor, Regisztráció

```
CREATE OR REPLACE TRIGGER ellenoriz_szuletesi_datum
BEFORE INSERT OR UPDATE ON FELHASZNALOK
FOR EACH ROW
BEGIN
    IF :NEW.szuletesi_datum > TRUNC(SYSDATE) THEN
        RAISE_APPLICATION_ERROR(-20010, 'A születési dátum nem lehet
        jövőbeli!');
    END IF;
END;
```

```
/
```

### Múltidőbeli járat tiltása

Adatbazis.sql, 897. sor, Járatok kezelése

```
CREATE OR REPLACE TRIGGER jarat_multido_ellenorzes
BEFORE INSERT OR UPDATE OF kiindulasi_idopont ON JARATOK
FOR EACH ROW
BEGIN
    IF :NEW.kiindulasi_idopont < SYSDATE THEN
        RAISE_APPLICATION_ERROR(-20107, 'A járat indulási időpontja nem lehet
        múltbeli időpont!');
    END IF;
END;
```

```
/
```

### Különböző városok járatok létrehozásánál

Adatbazis.sql, 906. sor, Járatok kezelése

```
CREATE OR REPLACE TRIGGER tr_check_different_cities
BEFORE INSERT OR UPDATE ON JARATOK
FOR EACH ROW
BEGIN
    IF :NEW.kiindulasi_hely = :NEW.erkezesi_hely THEN
        RAISE_APPLICATION_ERROR(-20002, 'A kiindulási hely és az érkezési hely
        nem lehet ugyanaz a város!');
    END IF;
END;
```

/

## Tárolt eljárások

### Jegyek száma emberenként

Adatbazis.sql, 843. sor, Kimutatások (TicketDAO 78 sor)

```
CREATE OR REPLACE PROCEDURE foglalas_statisztika (  
    p_cursor OUT SYS_REFCURSOR  
) AS  
BEGIN  
    OPEN p_cursor FOR  
        SELECT J.email as email, COUNT(*) AS darab  
        FROM JEGYEK J  
        GROUP BY J.email;
```

END;

/

### Keresés

Adatbazis.sql, 822. sor, Járatok keresése (FlightDAO 132 sor)

```
CREATE OR REPLACE PROCEDURE kereses(nap IN VARCHAR2, ki_hely IN NUMBER,  
be_hely IN NUMBER ,p_cursor OUT SYS_REFCURSOR) AS  
    cnt NUMBER;
```

BEGIN

```
    SELECT COUNT(*) INTO cnt  
    FROM JARATOK J  
    WHERE TRUNC(J.kiindulasi_idopont) = TRUNC(TO_DATE(nap,'YYYY-MM-DD  
HH24:MI:SS'))  
        AND J.kiindulasi_hely = ki_hely  
        AND J.erkezesi_hely = be_hely;  
    IF cnt=0 THEN  
        OPEN p_cursor FOR SELECT J1.id As first,J2.id as second FROM JARATOK  
        J1, JARATOK J2  
            WHERE J1.erkezesi_hely=J2.kiindulasi_hely AND  
            TRUNC(J1.kiindulasi_idopont)=TRUNC(J2.kiindulasi_idopont)  
            AND J1.erkezesi_idopont - TRUNC(J1.erkezesi_idopont) <  
            J2.kiindulasi_idopont - TRUNC(J2.kiindulasi_idopont)  
            AND TRUNC(J1.kiindulasi_idopont) = TRUNC(TO_DATE(nap,'YYYY-  
MM-DD HH24:MI:SS')) AND J1.kiindulasi_hely=ki_hely AND  
            J2.erkezesi_hely=be_hely;
```

Else

```

        OPEN p_cursor FOR SELECT J.id As first,J.id as second FROM JARATOK J
        WHERE TRUNC(J.kiindulasi_idopont) = TRUNC(TO_DATE(nap,'YYYY-MM-DD HH24:MI:SS')) AND J.kiindulasi_hely=ki_hely AND
        J.erkezesi_hely=be_hely;
    END if;
END;
/

```

## Legnépszerűbb járatok

Adatbázis.sql, 871. sor, Legnépszerűbb járatok (TicketDAO 82 sor)

```

CREATE OR REPLACE PROCEDURE legnepyszerubb_utvonalak(
    p_limit IN NUMBER,
    p_cursor OUT SYS_REFCURSOR
) AS
BEGIN
    OPEN p_cursor FOR
        SELECT V1.NEV AS KI, V2.NEV AS BE, COUNT(*) AS EMBEREK
        FROM VAROS V1, VAROS V2, JARATOK, FOGLALAS
        WHERE V1.ID = JARATOK.KIINDULASI_HELY
            AND V2.ID = JARATOK.ERKEZESI_HELY
            AND FOGLALAS.JARAT_ID = JARATOK.ID
        GROUP BY V1.NEV, V2.NEV
        ORDER BY EMBEREK DESC
        FETCH FIRST p_limit ROWS ONLY;
END;
/

```

Github link (ha gond van a beküldött zippel):

<https://github.com/vlrdex/adatb/tree/main>