

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. Első 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 számon 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 hozzáadása

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 funkciókhoz:
 - 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
 - Biztosítások tábla
 - Jegykategória tábla
 - Jegyek tábla
 - Modell tábla
 - Repülőgép tábla
 - Szállodák tábla
 - Város tábla
- Triggerek írása a specifikációban megjelölt funkciókhoz:
 - 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 funkciókhoz:
 - Jövőbeli születésidátum tiltása
- Tárolt eljárások/függvények a megjelölt funkciókhoz:
 - Jegyekszáma emberenként
 - Keresés
 - Legnépszerűbb járatok

Végső Bemutatás:

Bús Dávid:

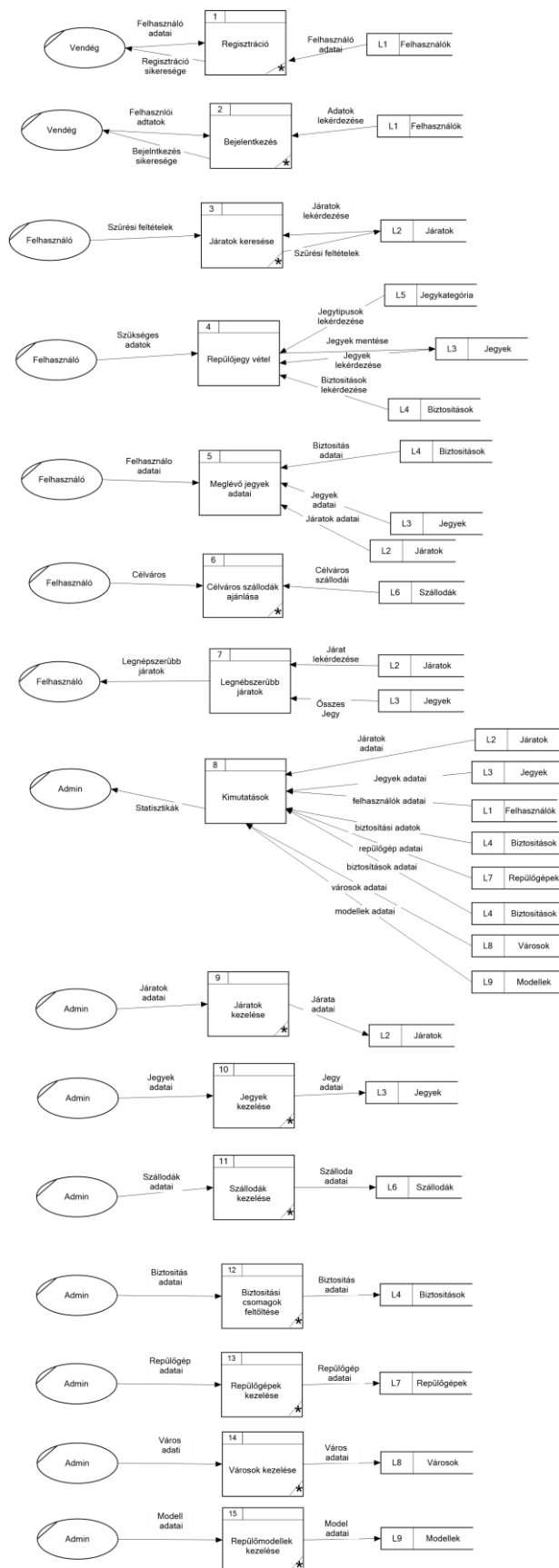
- Összetett lekérdezés:
 - Megjeleníti, hogy a foglalásokhoz milyen jegykategóriákat választottak

- Kilstázza, hogy az egyes járatokon hány szabad hely van még
Komlódi Tamás:

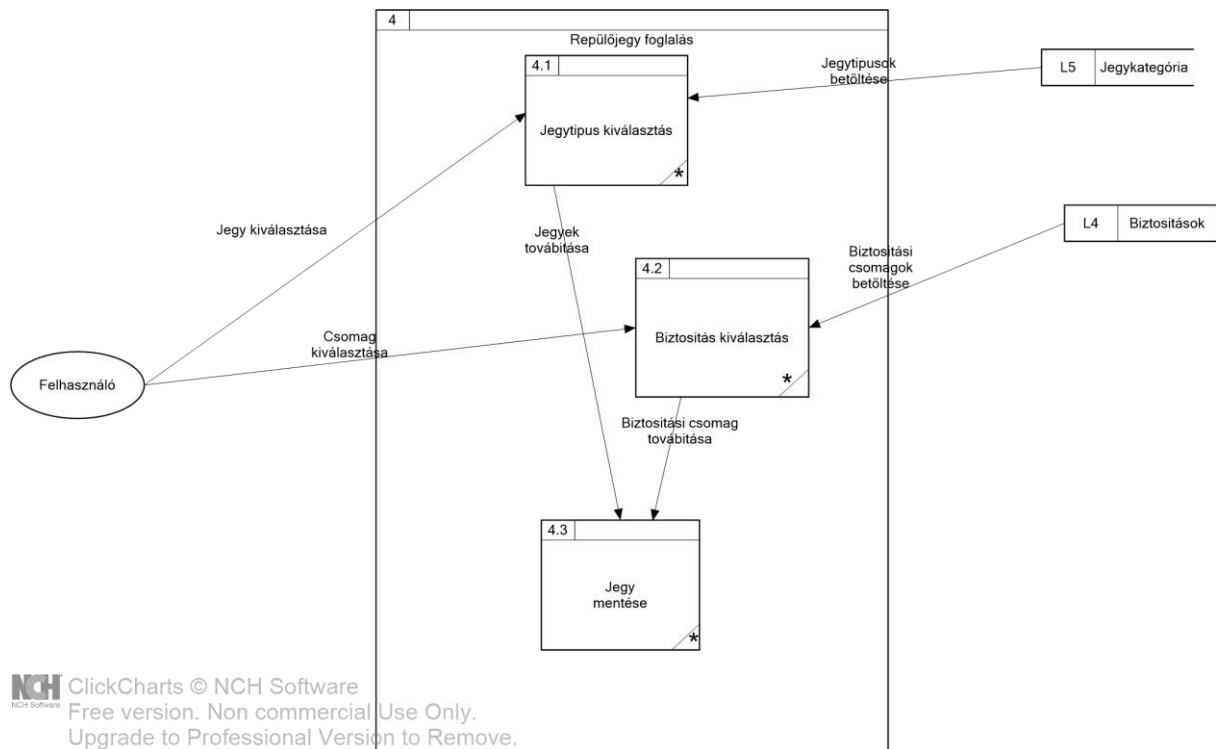
Tóth Péter Attila:

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

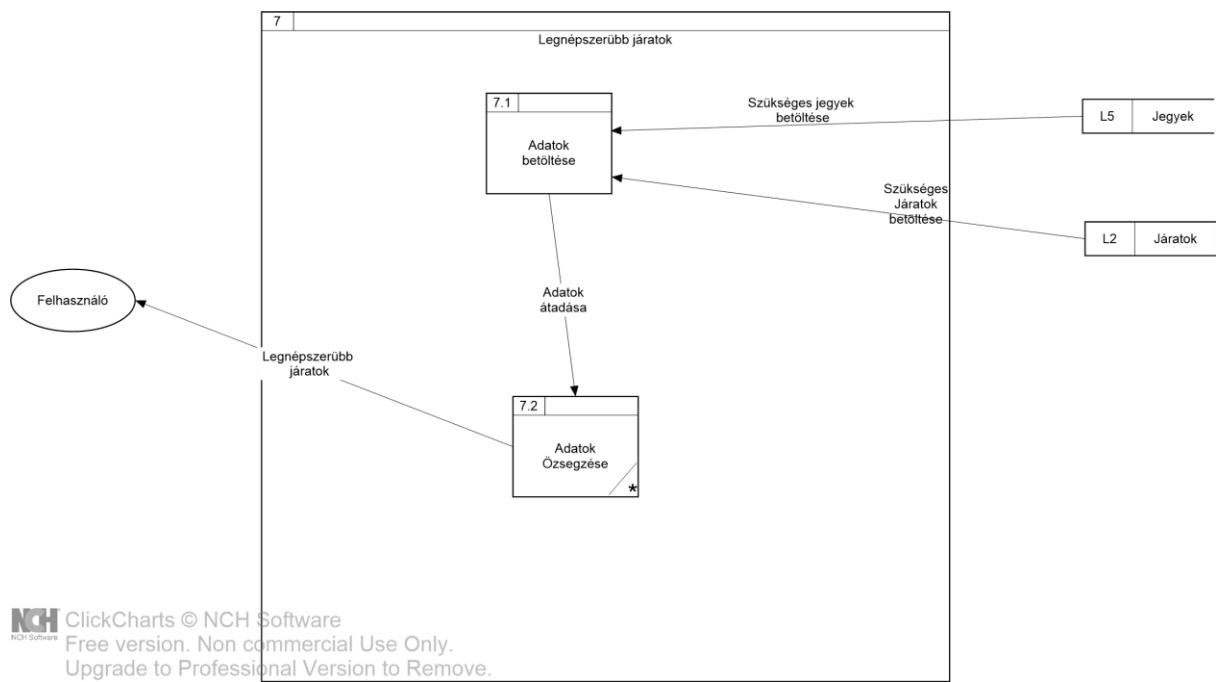
Logikai adatfolyam-diagramok



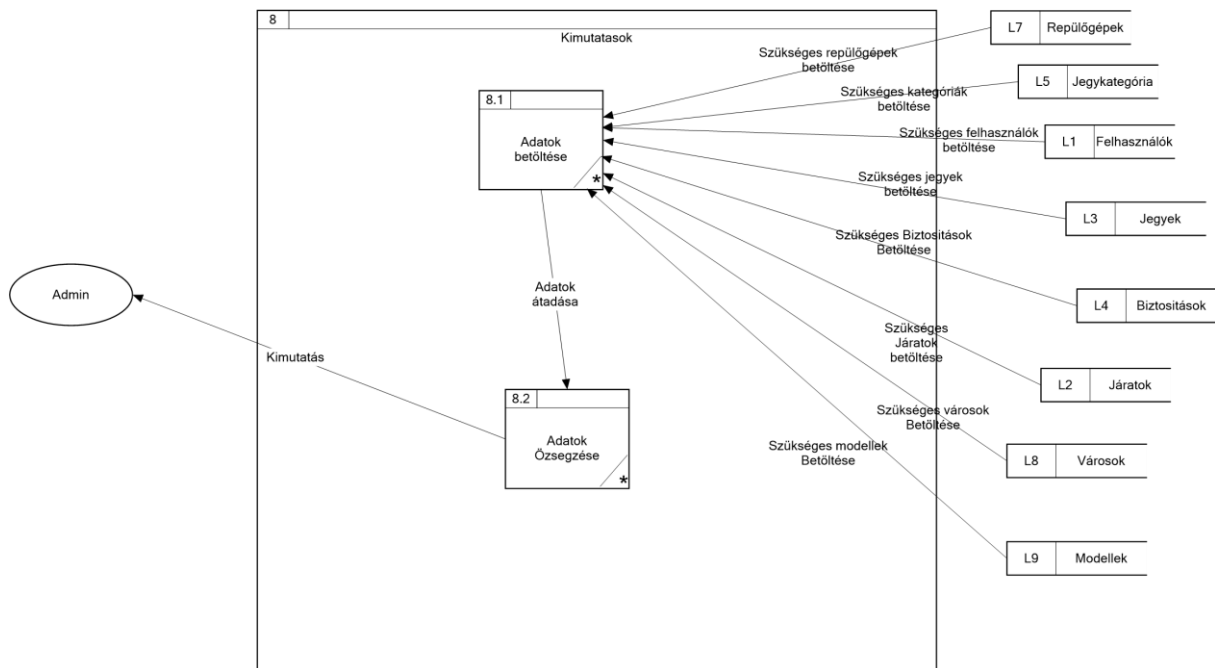
Első szint



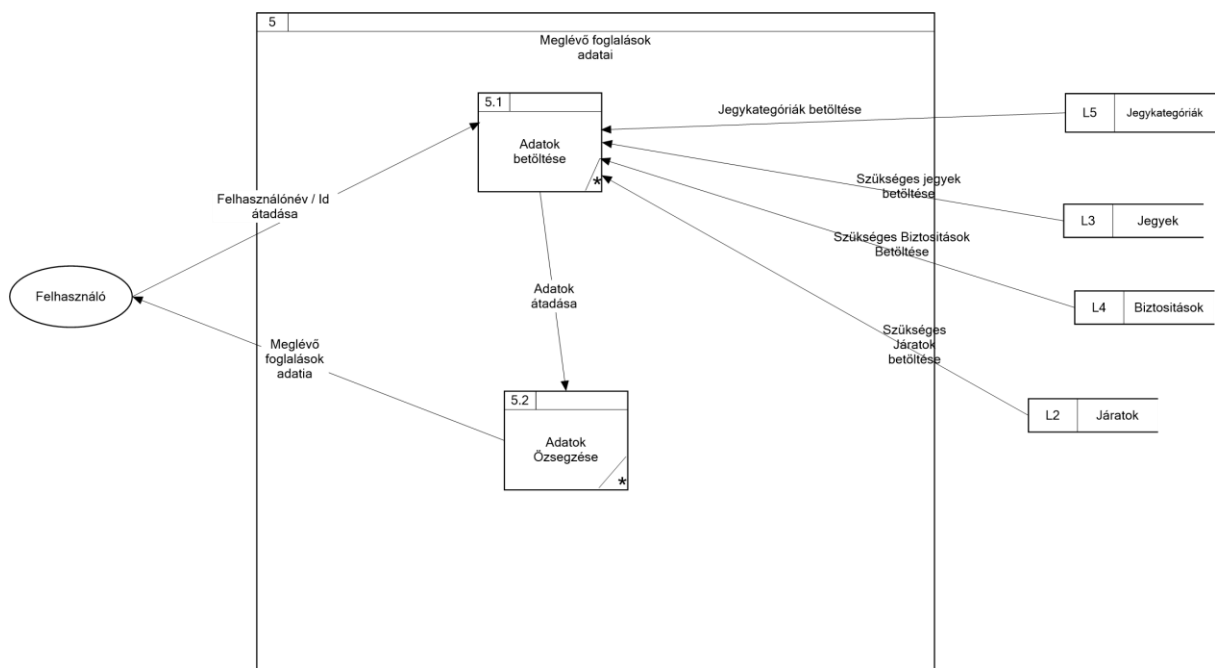
2. szint repülőjegy foglalás



2. szint Legnépszerűbb járatok

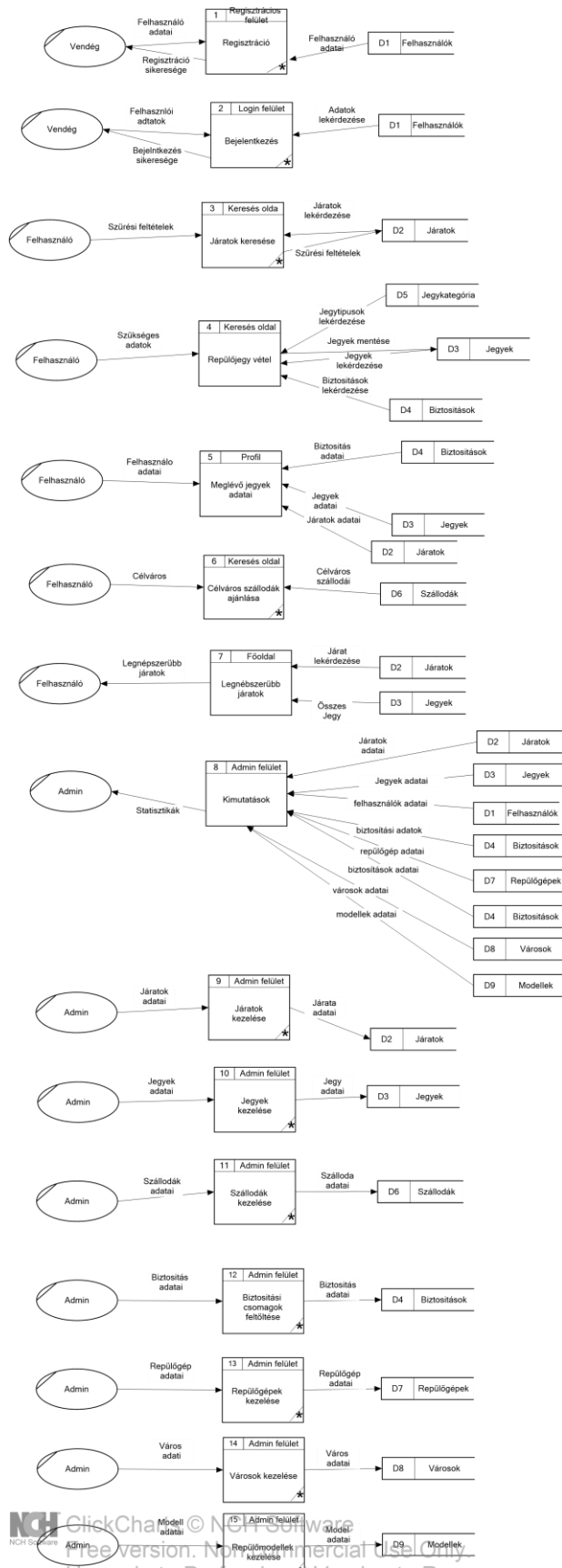


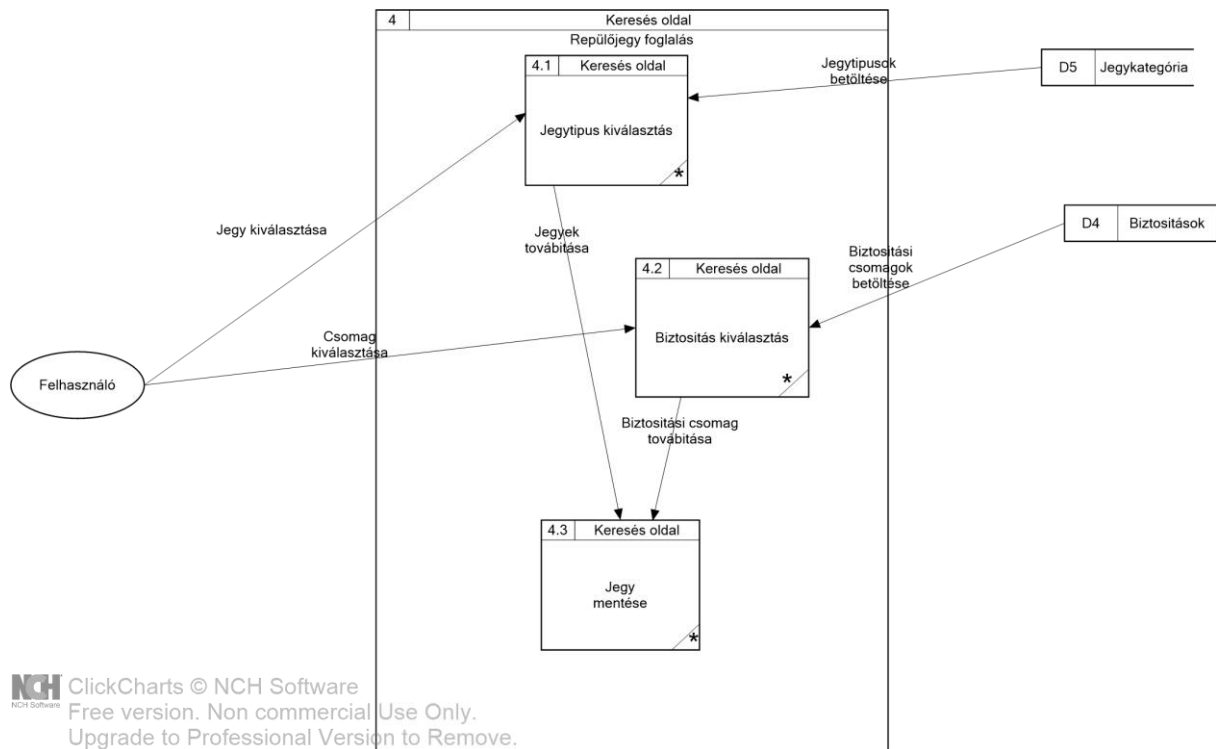
2. szint Kimutatások



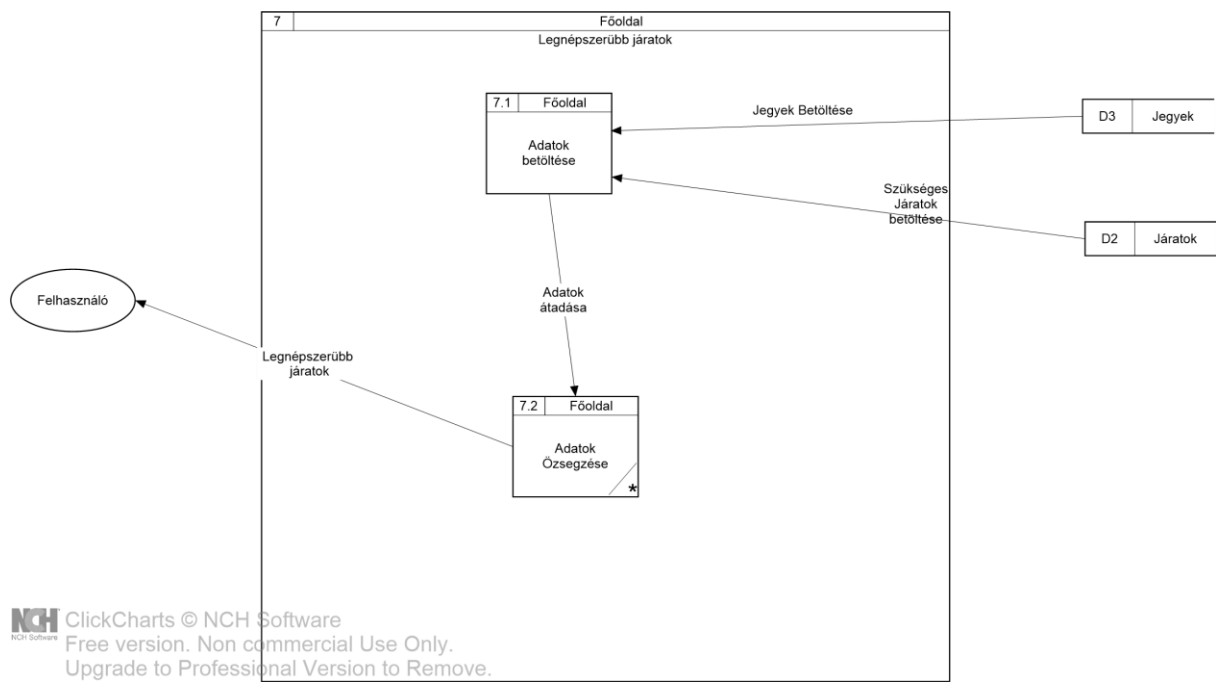
2. szint Meglévő foglalások

Fizikai adatfolyam-diagramok

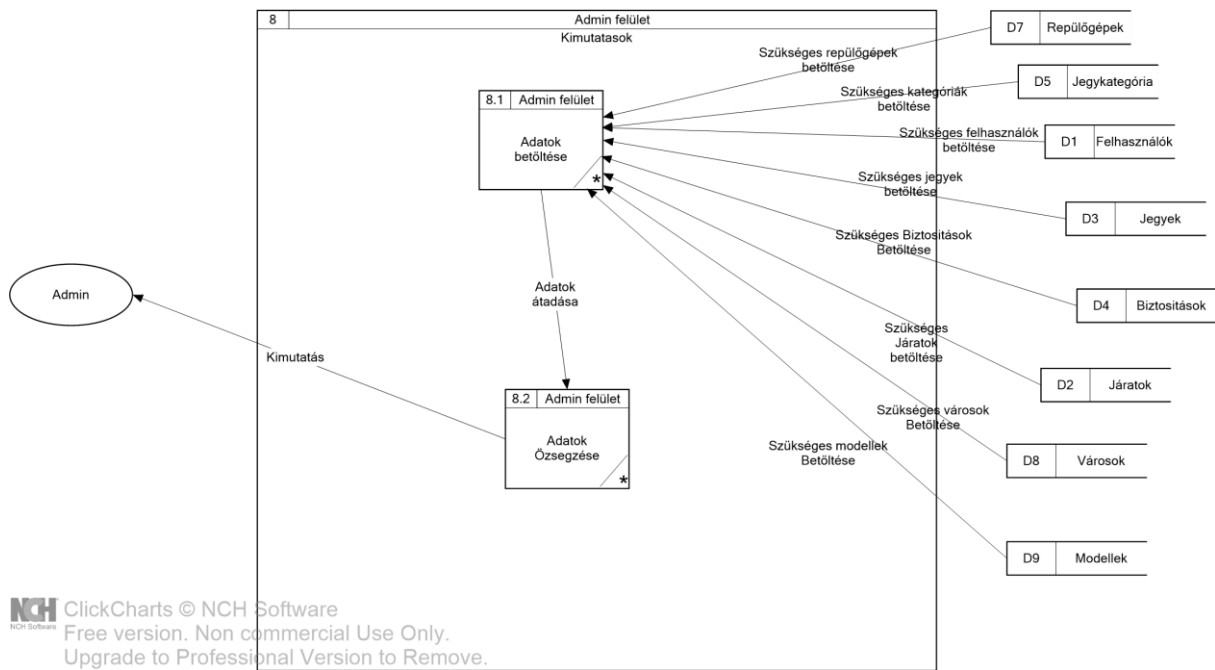




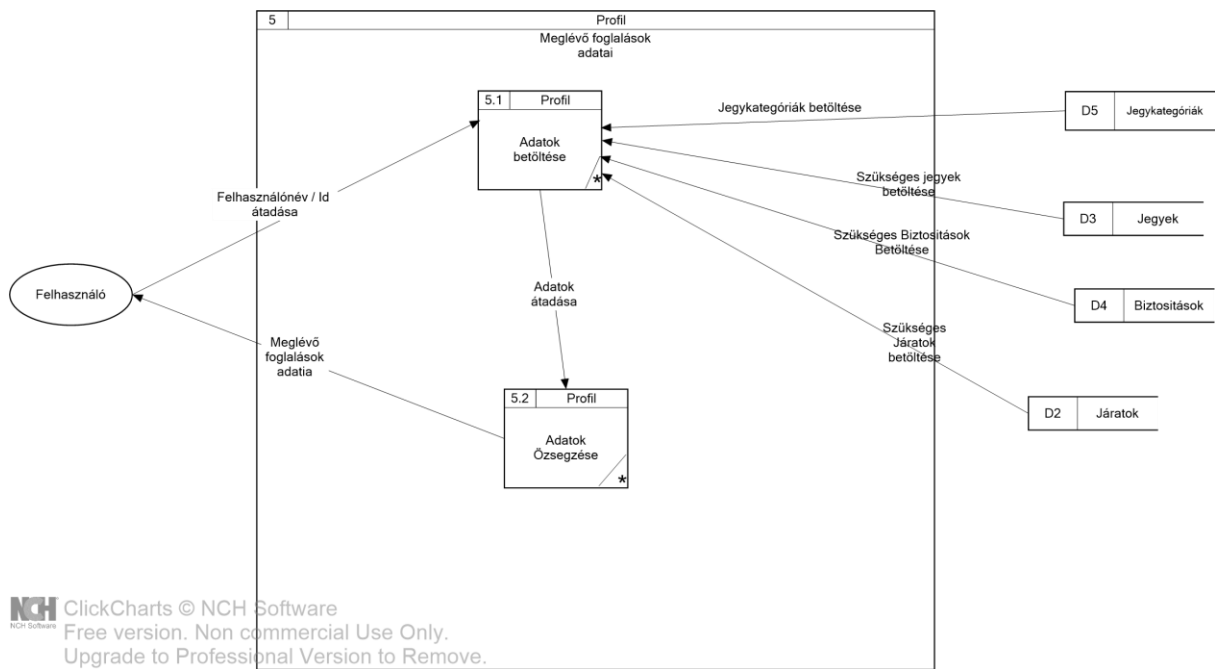
2. szint Repülőjegy foglalás



2. szint Legnépszerűbb járatok

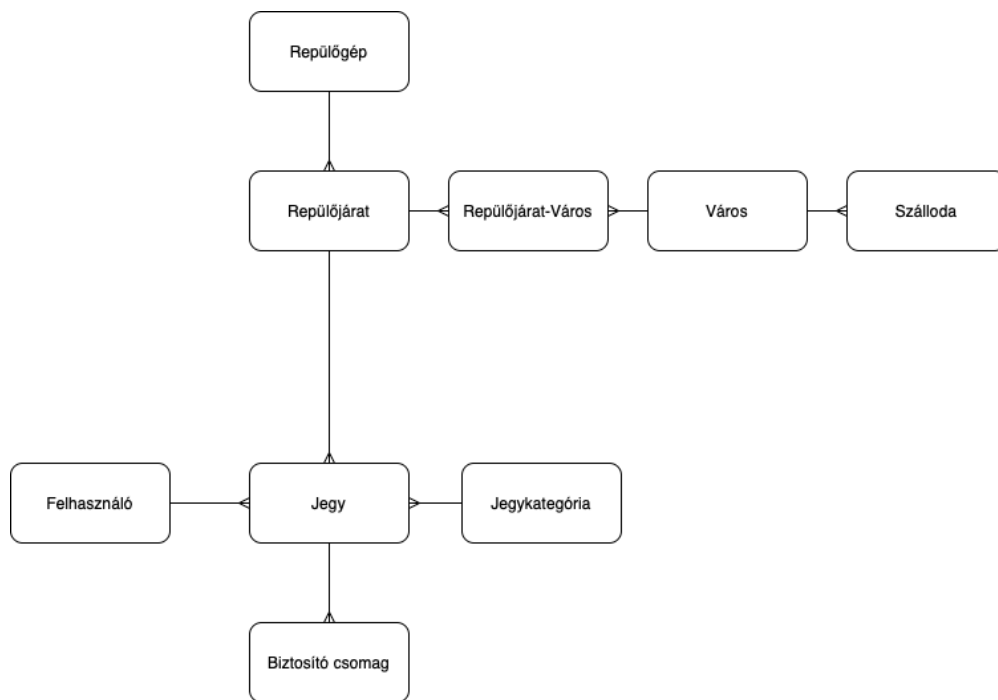


2. szint Kimutatások

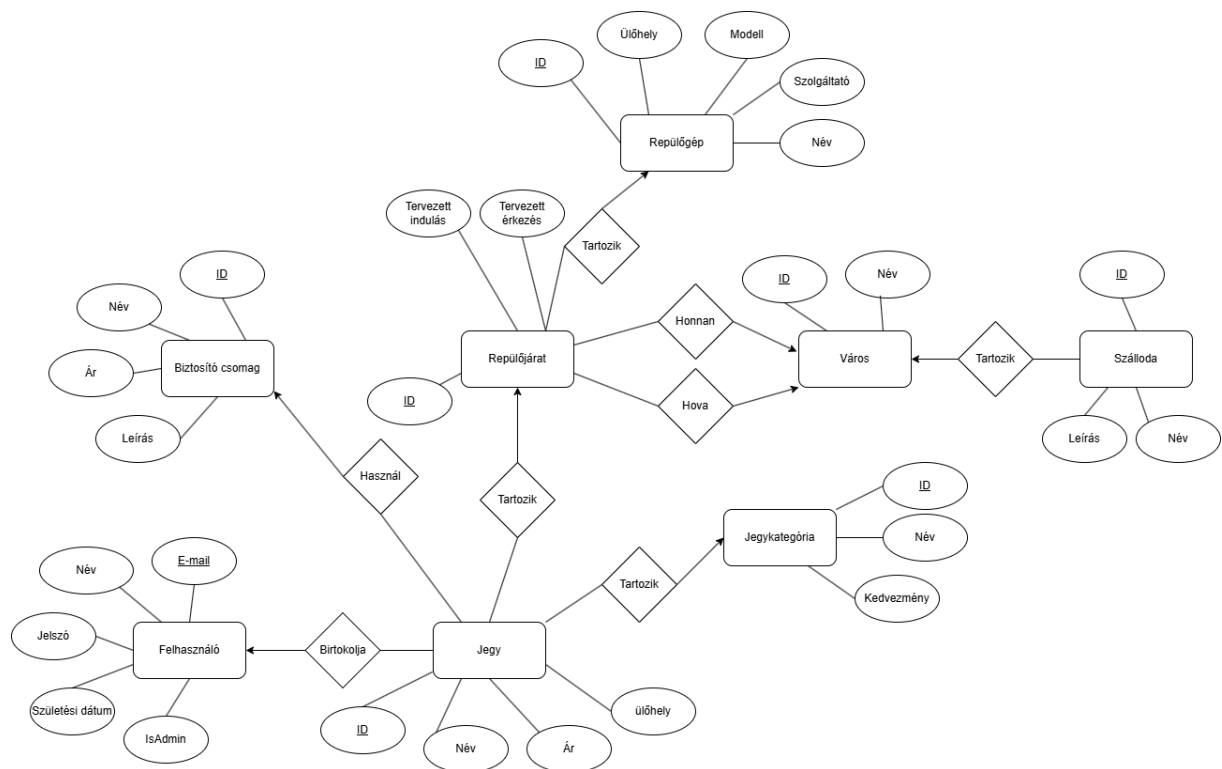


2. szint Foglalások

Egyedmodell



Egyed-kapcsolat diagram



Egyed-kapcsolat diagram leképezése relációs adatbázissémákká -

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 CASCADE
- **kiindulási hely** a **város** egyedben az **id**-re külső kulcs
 - ON UPDATE CASCADE
 - ON DELETE NO ACTION (RESTRICT)
- **érkezési hely** a **város** egyedben az **id**-re külső kulcs
 - ON UPDATE CASCADE
 - ON DELETE NO ACTION (RESTRICT)

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

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

- **járat id** a **járatok** egyedben az **id**-re külső kulcs
 - ON UPDATE CASCADE
 - ON DELETE CASCADE
- **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 NO ACTION (RESTRICT)

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).

JEGYEK

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

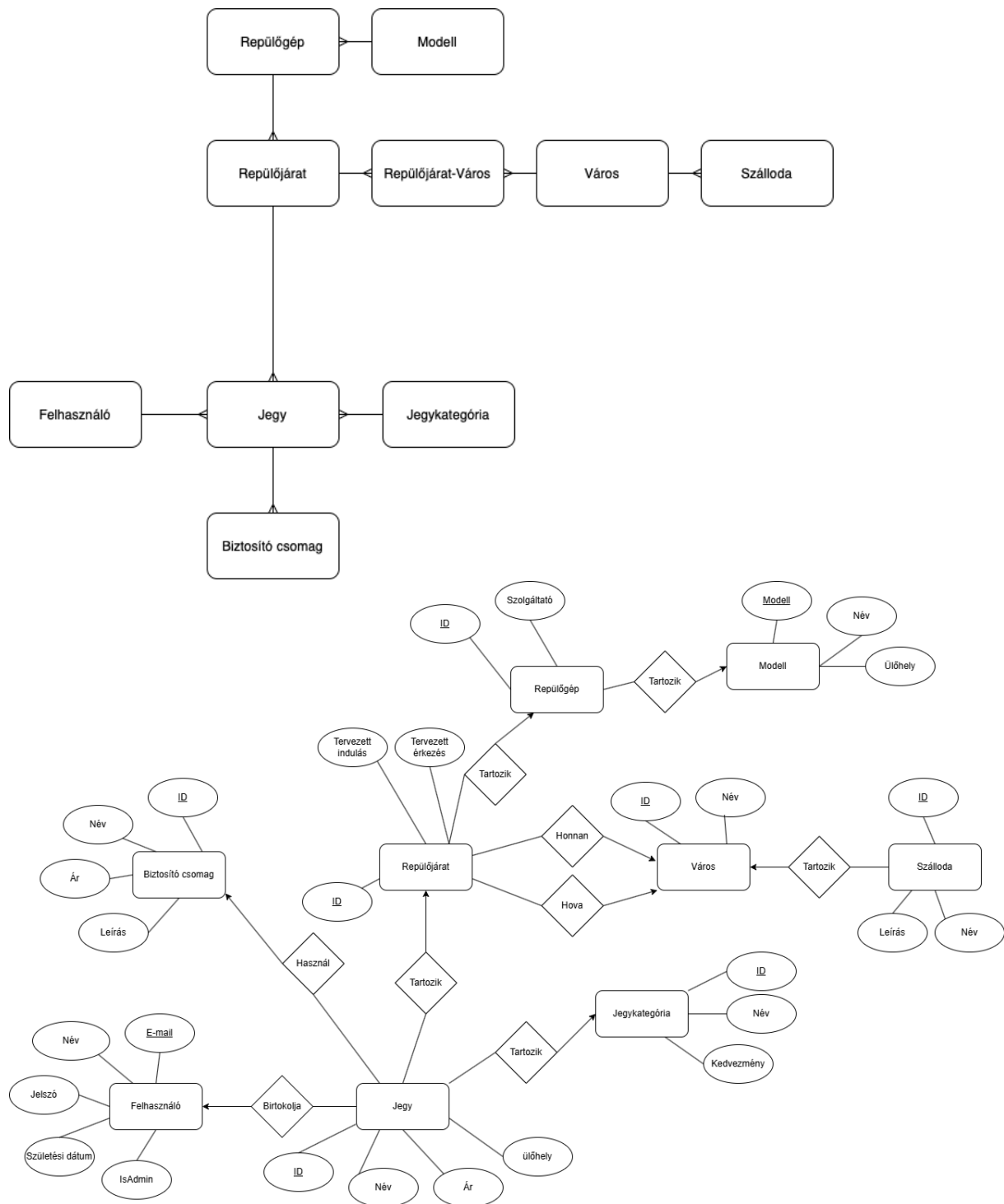
JEGYKATEGÓRIA

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

Változtatott tábla / egyedmodell / egyed-kapcsolat diagram

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 NO ACTION (RESTRICT)
- MODELL (modell, név, ülőhelyek száma)



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

-A név az UNIQUE kulcs.

JÁRATOK	PRIMARY KEY	FOREIGN KEY	NULL	DATATYPE
id	TRUE			INTEGER
kiindulási hely		TRUE		INTEGER
kiindulási időpont				DATE
érkezési hely		TRUE		INTEGER
érkezési időpont				DATE
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

-A név az UNIQUE kulcs.

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

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

-A név az UNIQUE kulcs.

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

-A név az UNIQUE kulcs.

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

-A város id és a név együtt UNIQUE kulcs.

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

-A név az UNIQUE kulcs.

Egyed-esemény mátrix

Egyed-esemény mátrix L=létrehozás O=olvasás M=módosítás T=törlés	Bejelentkezés	Regisztráció	Járatok keresése	Repülőjegy foglalás	Meglévő foglalások adatai	Célváros szállodák ajánlása	Legnépszerűbb járatok	Kimutatások	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)

Funkciónév: Regisztráció		Funkció azonosító: 1	
Típus: online, felhasználói			
Felhasználói szerepek: Jogosult: Vendég			
Funkció leírása: 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.			
Hibakezelés: 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.			
AFD-eljárások: 1			
Események: 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:

Funkciónév: Bejelentkezés		Funkció azonosító: 2	
Típus: online, felhasználói			
Felhasználói szerepek: vendég			
Funkció leírása: A vendég bejelentkeztetése.			
Hibakezelés: 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.			
AFD-eljárások: 2			
Események: 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			
Lekérdezések: 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:

Funkciónév: Járatok keresése		Funkció azonosító: 3	
Típus: online, felhasználói			
Felhasználói szerepek: vendég, admin, felhasználó			
Funkció leírása: Járatok keresése, az adott paraméterekkel. Miután beállítottuk, hogy hova, mikor és milyen módon szeretnénk utazni, válaszként egy listát kapunk az adott ideális járatokból.			
Hibakezelés: 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!”			
AFD-eljárások: 3			
Események: 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:

Funkciónév: Repülőjegy foglalása		Funkció azonosító: 4	
Típus: online, felhasználói			
Felhasználói szerepek: admin, felhasználó			
Funkció leírása: 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.			
Hibakezelés:			
AFD-eljárások: 4			
Események: Minden jegy vásárlásnál		Esemény gyakoriság: Minden jegyvásárlásnál	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.5			
Tömegszerűség: Naponta többször			
Kapcsolódó funkciók:			
Lekérdezések: Minden adatot lekérünk az adott jegyhez		Lekérdezés gyakorisága: Minden vásárlásnál	
Közös feldolgozás: Összes pontos adat lekérése az adott kiválasztott jegyhez			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Jegy mentése	Célérték: 3 mp	Tartomány: 4-6 mp	Megjegyzések:

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

Funkciónév: Célváros szállodák ajánlása		Funkció azonosító: 6	
Típus: online, felhasználói			
Felhasználói szerepek: vendég, felhasználó, admin			
Funkció leírása: Jegy foglaláskor a program felajánl szállodákat az adott célvárosban, amik közül lehet választani, hogy hol szállna meg.			
Hibakezelés:			
AFD-eljárások: 6			
Események: 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:

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

Funkciónév: Kimutatások		Funkció azonosító: 8	
Típus: online, admin, karbantartó			
Felhasználói szerepek: admin			
Funkció leírása: Havi/éves kimutatási statisztikák az admin felületre. Bevételek kiszámítása. Gyakori járatok és egyéb hasznos információ.			
Hibakezelés: Helytelen szűrés feltételnél hiba üzenet dob, hogy „Nem létezik adott statisztika!”. Kevés adat esetében is ez az üzenet jelenik meg.			
AFD-eljárások: 8			
Események: 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:

Funkciónév: Járatok feltöltése		Funkció azonosító: 9	
Típus: online, admin, karbantartó			
Felhasználói szerepek: admin			
Funkció leírása: Az admin feltudja tölteni az adatbázist új járatokkal. Ezeket tudja módosítani vagy törölni is.			
Hibakezelés: 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.			
AFD-eljárások: 9			
Események: Járatok kezelése		Esemény gyakoriság: Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.1, 2.14			
Tömegszerűség: Hetente			
Kapcsolódó funkciók:			
Lekérdezések: A repülőgépekre és városokra		Lekérdezés gyakorisága: Járatok kezelésénél	
Közös feldolgozás: Repülőgép és város adatok			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Járatok kezelése	Célérték: 4 mp	Tartomány: 5-7 mp	Megjegyzések:

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

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

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

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

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

Funkciónév: Repülőmodellek kezelése		Funkció azonosító: 15	
Típus: online, admin, karbantartói			
Felhasználói szerepek: admin			
Funkció leírása: Az admin tudja kezelni a modelleket. Hozzá tudni adni, módosítani vagy törölni.			
Hibakezelés: Hibás adatnál „Hibás adattípus!” üzenet fog megjelenni. Üres mezők esetén „Kérem töltsön ki minden mezőt!”.			
AFD-eljárások: 15			
Események: Repülőmodellek kezelése kezelése		Esemény gyakoriság: Admin függő	
I/O leírások:			
I/O szerkezetek:			
Követelménykatalógus hivatkozás: 2.17			
Tömegszerűség: Havonta 1x			
Kapcsolódó funkciók:			
Lekérdezések: Modellek adatai		Lekérdezés gyakorisága: Minden kezelésnél	
Közös feldolgozás: Repülőmodellek lekérdezése			
Dialógusnevek:			
Szolgáltatási szint követelményei:			
Leírása: Repülőmodellek kezelése	Célérték: 4 mp	Tartomány: 6-8 mp	Megjegyzések:

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(JEGYEK.nev) AS COUNT
```

```

FROM JARATOK
JOIN VAROS BE ON JARATOK.KIINDULASI_HELY = BE.ID
JOIN VAROS KI ON JARATOK.ERKEZESI_HELY = KI.ID
JOIN JEGYEK ON JARATOK.ID = JEGYEK.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(JEGYEK.nev) AS COUNT
FROM JARATOK
JOIN VAROS BE ON JARATOK.KIINDULASI_HELY = BE.ID
JOIN VAROS KI ON JARATOK.ERKEZESI_HELY = KI.ID
JOIN JEGYEK ON JARATOK.ID = JEGYEK.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 REPULOGEP r ON j.repulo_id = r.id
      LEFT JOIN JEGYEK jegy ON j.id = jegy.jarat_id
      LEFT JOIN JEGYKATEGORIA jk ON jegy.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
```

Utások 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(jar.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 REPULOGEPR
    JOIN JARATOK j ON r.id = j.repulo_id
    LEFT JOIN JEGYEK jegy ON j.id = jegy.jarat_id
    LEFT JOIN JEGYKATEGORIA jk ON jegy.jegykatgoria_id = jk.id
GROUP BY r.szolgáltato
ORDER BY jegy_bevetel DESC

```

Napi bevétel és eladott jegyszám

FlightDAO 150.sor, Kimutatások funkció

```

SELECT
    TRUNC(j.kiindulasi_idopont) AS SZOLGALTATO,
    COUNT(f.nev) AS OSSZES_ELADOTT_JEGY,
    SUM(j.ar) AS JEGY_BEVETEL

```



```

FROM
    JEGYEK 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.szolgaltato
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 168. sor, getTicketStat(), Kimutatások

```

SELECT
    jk.nev AS jegykategoria,
    jk.kedvezmeny AS kedvezmeny_szazalek,
    COUNT(f.nev) AS foglalasok_szama,
    ROUND(AVG(j.ar * (1 - jk.kedvezmeny/100)), 2) AS atlagos_fizetett_ar
FROM JEGYEK 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 181. 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

ADAT.sql, 642. 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

ADAT.sql, 703. sor, Repülőjegy foglalás

```
CREATE OR REPLACE TRIGGER foglalt_ulohely
BEFORE INSERT
ON JEGYEK
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
```

```

SELECT COUNT(*) INTO v_count
FROM JEGYEK J
WHERE JARAT_ID = :NEW.JARAT_ID
AND J.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

ADAT.sql, 737. 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

ADAT.sql, 746. 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

ADAT.sql, 755. 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

ADAT.sql, 692. 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

ADAT.sql, 671. sor, Járatok keresése (FlightDAO 130 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

ADAT.sql, 651. 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, JEGYEK
        WHERE V1.ID = JARATOK.KIINDULASI_HELY
            AND V2.ID = JARATOK.ERKEZESI_HELY
            AND JEGYEK.JARAT_ID = JARATOK.ID
        GROUP BY V1.NEV, V2.NEV
        ORDER BY EMBEREK DESC
        FETCH FIRST p_limit ROWS ONLY;
END;
/

```

Szoftver környezet:

- **Java Development Kit (JDK)** – 17 vagy újabb verzió
- **Spring Boot** – 3.x verzió (Maven)
- **Oracle Database** – 12c
- **IntelliJ IDEA** – fejlesztői környezet (Community vagy Ultimate)
- **Google Chrome** – frontend megjelenítéshez

Github link (ha gond van a beküldött zippel):

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