|  |  |
| --- | --- |
| **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 feladatleírás, 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

## 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

1. Mérföldkő

Komlódi Tamás

* Adatbázist létrehozó szkript

1. 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

1. 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
* Biztositasok 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 funkciókhoz:
* Jegyekszáma emberenként
* Keresés
* Legnépszerűbb járatok

Végső Bemutatás:

Bús Dávid:

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

Komlódi Tamás:

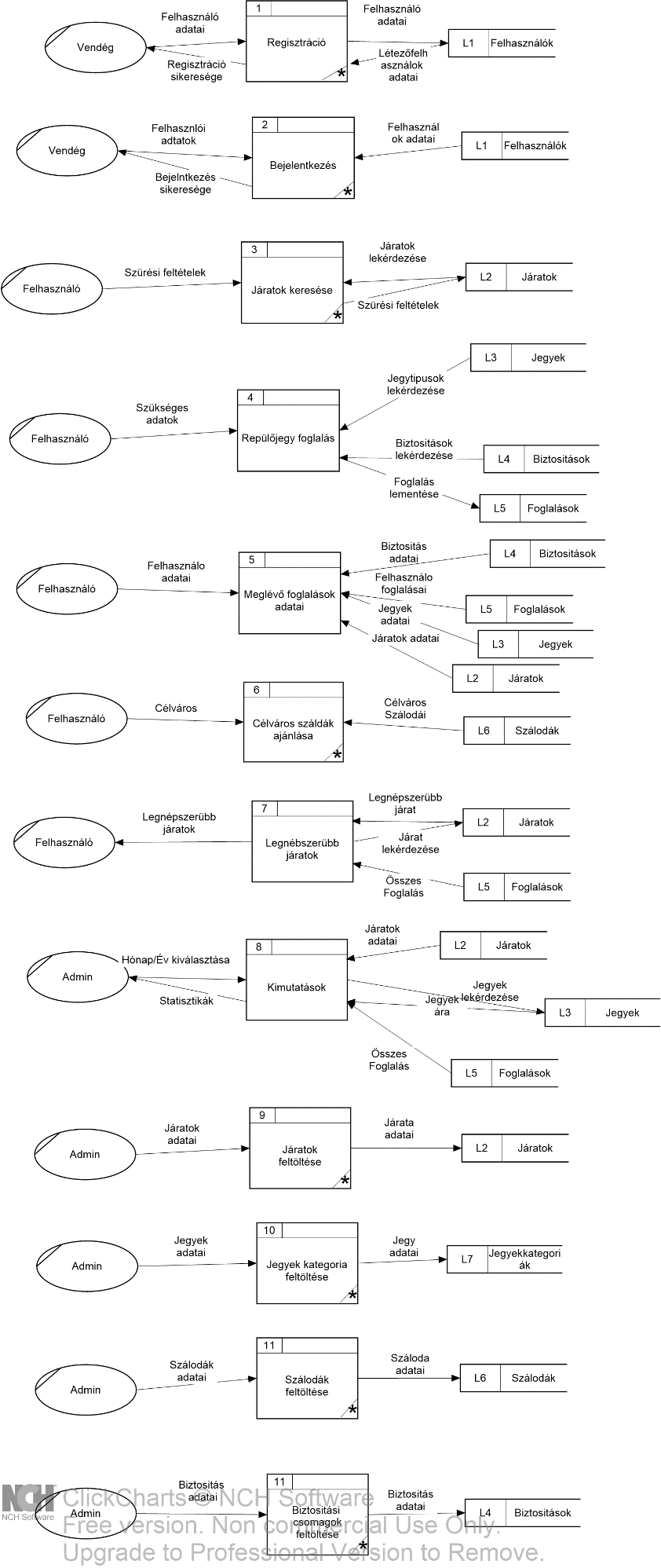
Tóth Péter Attila:

* Összetett lekérdezés:

- Napi bevétel és eladott jegyszám

- Az átlagnál drágább járatok

## Logikai adatfolyam-diagramok



Első szint

A képen diagram, Tervrajz, szöveg, sor látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint repülőjegy foglalás

A képen diagram, szöveg, sor, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint Legnépszerűbb járatok

A képen diagram, sor, Párhuzamos, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint Kimutatások

A képen diagram, sor, Párhuzamos, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint Meglévő foglalások

A képen szöveg, diagram, Párhuzamos, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. szint Legnépszerűbb adatok betöltése

A képen szöveg, diagram, Tervrajz, Műszaki rajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

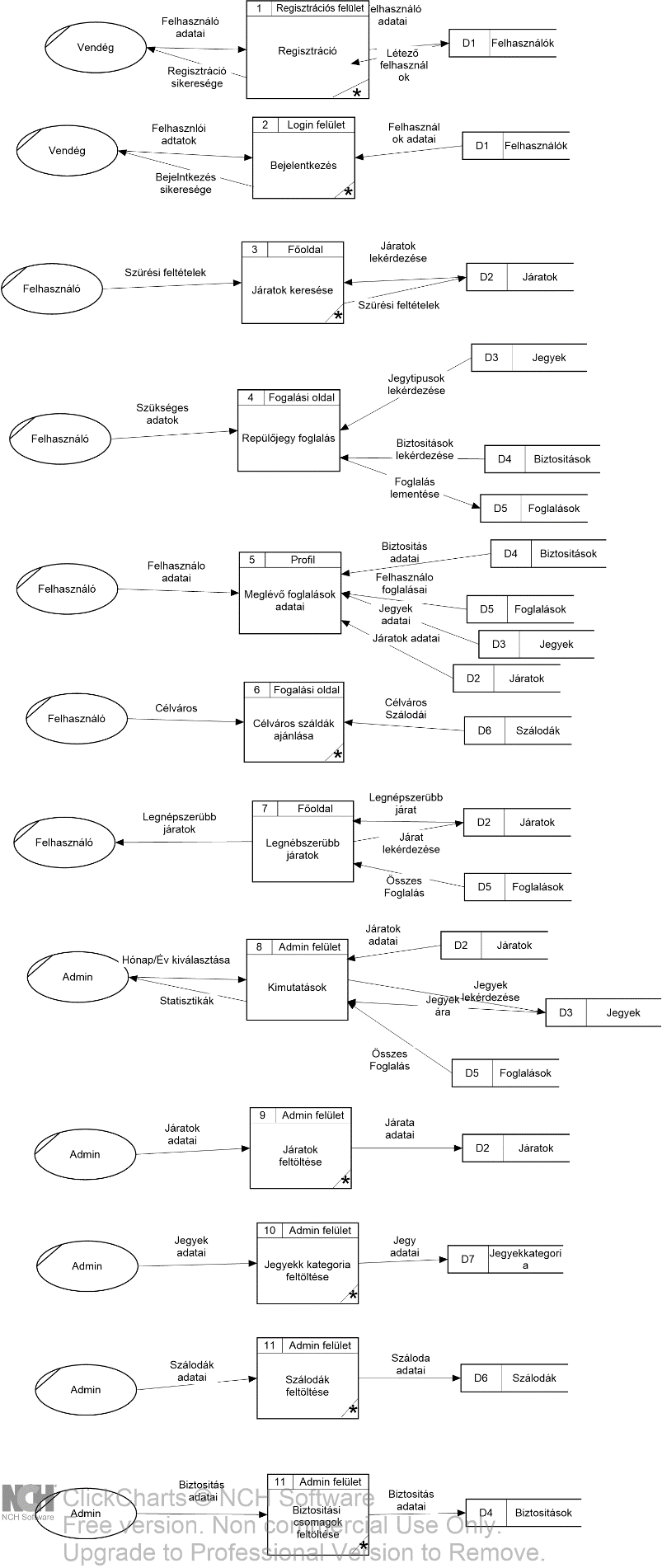
3. szint kimutatás adatok betöltése

A képen szöveg, diagram, Tervrajz, Műszaki rajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. szint Adatok betöltése

## Fizikai adatfolyam-diagramok



. szint

A képen diagram, képernyőkép, Tervrajz, Műszaki rajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

. szint Repülőjegy foglalás

A képen diagram, szöveg, sor, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint Legnépszerűbb járatok

A képen diagram, sor, Párhuzamos, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint Kimutatások

A képen diagram, sor, Párhuzamos, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

2. szint Foglalások

A képen szöveg, diagram, Párhuzamos, Tervrajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

. szint Adatok

A képen szöveg, diagram, Tervrajz, Műszaki rajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. szint Kimutatás adatok

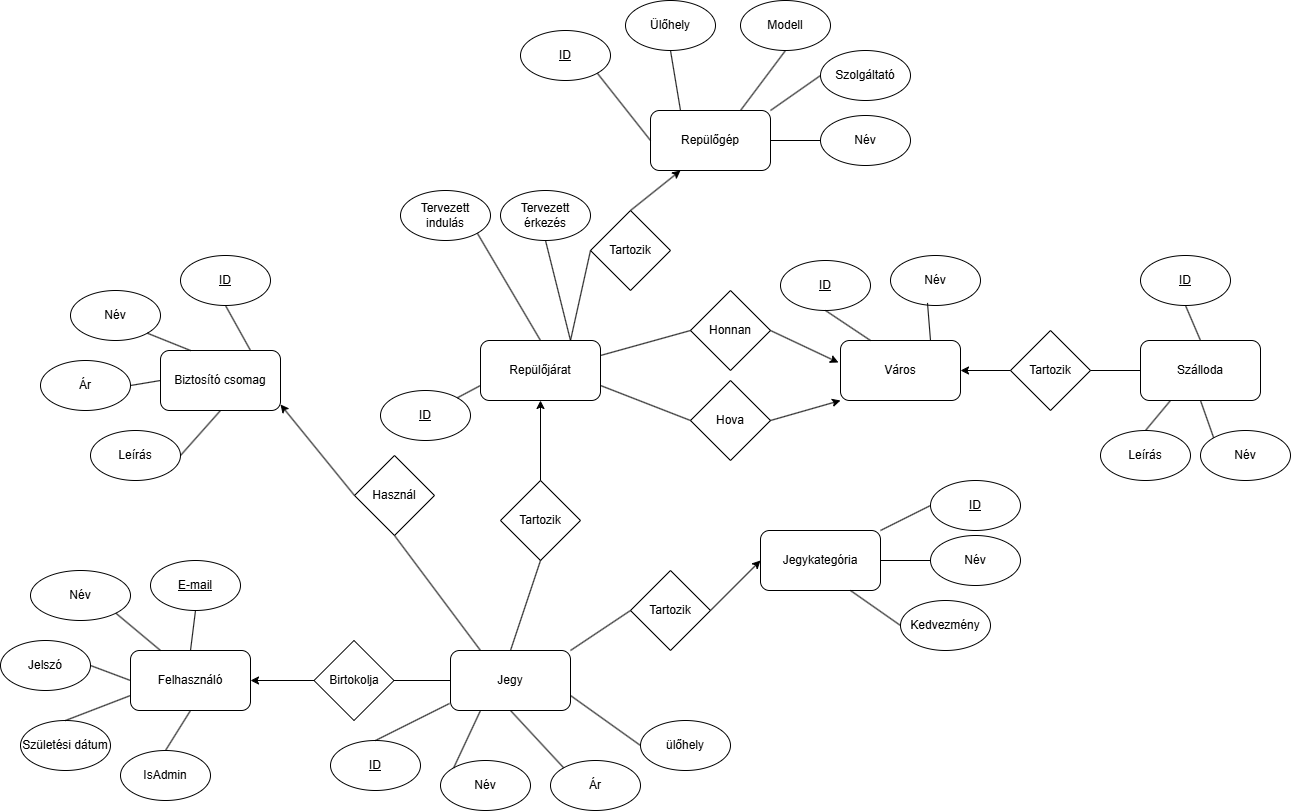
A képen diagram, szöveg, Tervrajz, Műszaki rajz látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

3. szint Adatok

## 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 CASCADE
* **érkezési hely** a **város** egyedben az **id**-re külső kulcs
* ON UPDATE CASCADE
* ON DELETE CASCADE

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

JEGYEK (*járat id*, ülőhely, *biztositás 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 SET CASCADE

VÁROS (id, név)

Alulról felfelé történő elemzés (funkcionális függőségek alapján)

FELHASZNÁLÓK

* {email} -> {név, jelszó, születési dátum, admin}
* Feltételezük, 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}
* Feltételezük 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).

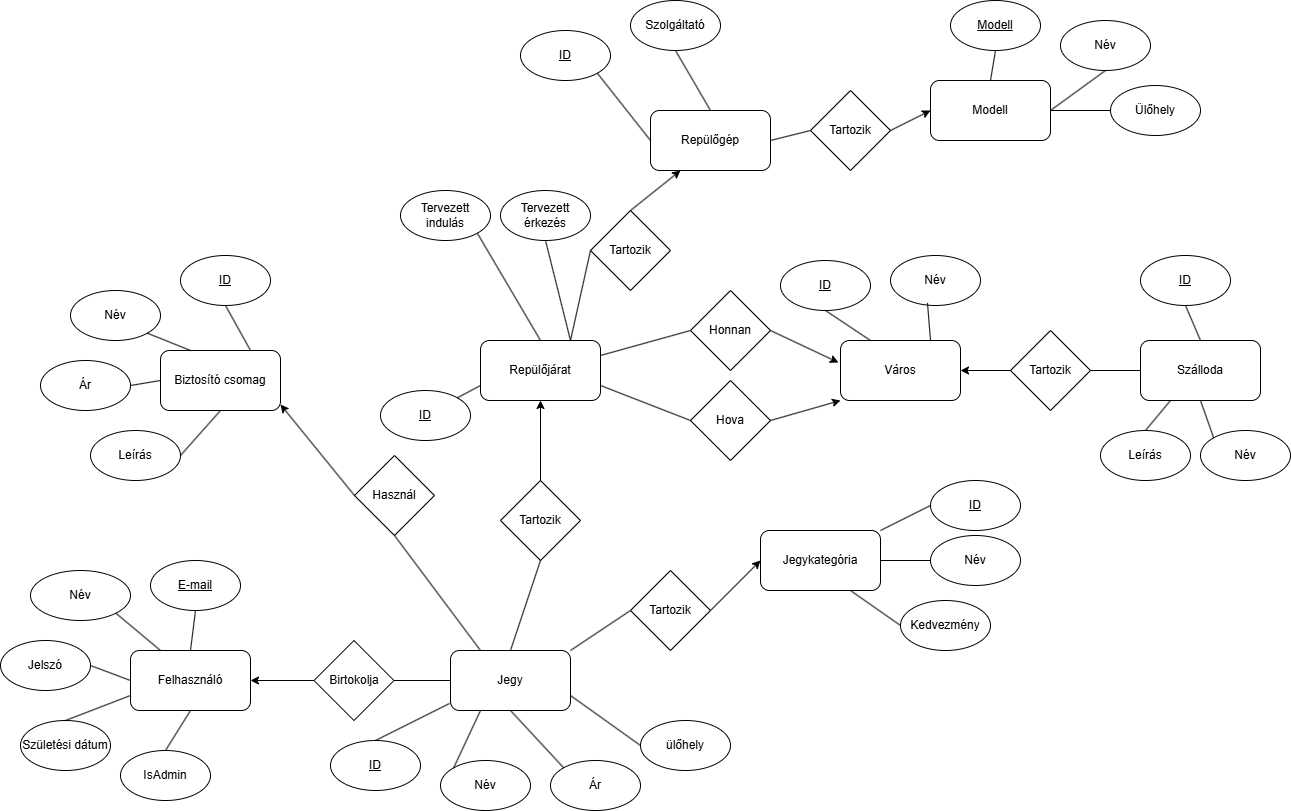
JEGYKATEGÓRIA

* {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 CASCADE
* MODELL (modell, név, ülőhelyek száma)



A képen képernyőkép, szöveg, Betűtípus, tervezés látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

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

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

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

## 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 | Biztositások 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 |  | LMT |  |  |  |  |  |
| Modell |  |  |  | O |  |  |  | O |  |  |  |  |  |  | LMT |

## Szerep-funkció mátrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Szerep-funkció  mátrix | Bejelentkezés | Regisztráció | Járatok keresése | Repülőjegy foglalás | Meglévő foglalások adatai | Célváros szállodák ajánlása | Legnépszerűbb járatok | Kimutatások | Járatok kezelése | Jegykategóriák kezelése | Szállodák kezelése | Biztositások kezelése | Repülőgépek kezelése | Városok kezelése | Repülőmodellek kezelése |
| Vendég | x | x | x |  |  | x | x |  |  |  |  |  |  |  |  |
| Felhasználó |  |  | x | x | x | x | x |  |  |  |  |  |  |  |  |
| Admin |  |  | x | x | x | x | x | x | x | x | x | x | x | x | x |

## 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 megkell 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 | | | |
| Dialógusnevek: | | | |
| 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 megnyitásakkor |
| 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 | | | |
| Dialógusnevek: | | | |
| 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: | | | |
| Dialógusnevek: | | | |
| 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:** 11 | | | |
| **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:** 11 | | | |
| **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: Szállodák lekérdezése | | | |
| Dialógusnevek: | | | |
| Szolgáltatási szint követelményei: | | | |
| Leírása:  Városok kezelése | Célérték:  4 mp | Tartomány:  6-8 mp | Megjegyzések: |

|  |  |  |  |
| --- | --- | --- | --- |
| **Funkciónév:**  Repülőmodellek kezelése | | | **Funkció azonosító:**  14 |
| **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:** 11 | | | |
| **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 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.jegykategoria\_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(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.szolgaltato, 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 JEGYKATEGORIA jk ON jegy.jegykategoria\_id = jk.id

GROUP BY r.szolgaltato

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

REPULOGEP 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, 679. 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;

/

### Foglalalt ülőhelyek tiltása

ADAT.sql, 740. 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, 774. 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, 783. 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 létrehozássánál

ADAT.sql, 792. 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, 729. 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, 708. 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, 757. sor, Legnépszerűbb járatok (TicketDAO 82 sor)

CREATE OR REPLACE PROCEDURE legnepszerubb\_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>