Adatbázis alapú rendszerek

2024/2025/2

IB152L-8

Csütörtök 8:00-10:00

**Vasútmenetrend**

**Készítette:**

Domokos Ádám

Hajagos János

Sipka Ádám Kadosa

## Munka felosztása

Domokos Ádám

* 1. Mérföldkő
  + Egyedmodell
  + Egyed-kapcsolat diagram
  + Relációsémák
    - Normalizálás
  + Táblák leírása
* 2. Mérföldkő
  + Járat tábla
  + Vonat tábla
  + Csatlakozás tábla
  + Állomás tábla
  + Rekordok a táblákhoz
* 3. Mérföldkő
  + Járatok kezelése (felvitel, módosítás, törlés)
  + Vonatok kezelése (felvitel, módosítás, törlés)
  + Csatlakozások kezelése (felvitel, módosítás, törlés)
  + Állomások kezelése (felvitel, módosítás, törlés)
  + Jegyek kezelése (felvitel, módosítás, törlés)
* 4. Mérföldkő
  + Járatok alapadatainak lekérdezése
  + Vonatok alapadatainak lekérdezése
  + Csatlakozások alapadatainak lekérdezése
  + Állomások alapadatainak lekérdezése
  + Jegyek alapadatainak lekérdezése
  + Kedvezményeket kezelő trigger

Hajagos János

* 1. Mérföldkő
  + Egyed-esemény mátrix
  + Szerep-funkció mátrix
  + Funkció-meghatározás
* 2. Mérföldkő
  + Felhasználó tábla
  + Kedvezmény tábla
  + Alkalmazott tábla
  + Szabadság tábla
  + Rekordok a táblákhoz
* 3. Mérföldkő
  + Felhasználók kezelése (felvitel, módosítás, törlés)
  + Kedvezmények kezelése (felvitel, módosítás, törlés)
  + Alkalmazottak kezelése (felvitel, módosítás, törlés)
  + Szabadságok kezelése (felvitel, módosítás, törlés)
  + Munkabeosztás kezelése (felvitel, módosítás, törlés)
* 4. Mérföldkő
  + Felhasználók alapadatainak lekérdezése
  + Kedvezmények alapadatainak lekérdezése
  + Alkalmazottak alapadatainak lekérdezése
  + Szabadságok alapadatainak lekérdezése
  + Munkabeosztás alapadatainak lekérdezése
  + Beosztások ütközésénél trigger
  + Alkalmazotti bérek kiszámítása (Lekérdezés)
  + Kalauzok, jegypénztárosok beosztása
  + Új szabadság rögzítése (Tárolt eljárás)
  + Bevétel – kiadás = nyereség (Éves kimutatás)(összetett lekérdezés)
  + Szabadságokat ellenőrző trigger

Sipka Ádám Kadosa

* 1. Mérföldkő
  + Fizikai adatfolyam-diagrammok
  + Logikai adatfolyam-diagrammok
  + Követelménykatalógus, részletes feladatleírás, specifikáció
* 2. Mérföldkő
  + Dokumentáció aktualizálása
  + Vásárlás tábla
  + Munkabeosztás tábla
  + Jegyek tábla
  + Rekordok a táblákhoz
* 3. Mérföldkő
  + Dokumentáció aktualizálása
  + Regisztrációs űrlap
  + Bejelentkezési űrlap
* 4. Mérföldkő
  + Dokumentáció aktualizálása
  + Menetrendet összerakó funkció
  + Összetett lekérdezések

## Feladat szöveges leírása

## A feladat egy olyan alkalmazást foglal magába, amelyen keresztül egy vasúttársaság menetrendjét lehet megtekinteni, illetve jegyet foglalni járataira. Továbbá, a társaság adminisztrátorai az alkalmazásban tudják kezelni az alkalmazottaik (kalauzok, jegypénztárosok) beosztását, és többfajta statisztikát és adatot tudnak lekérdezni (például a jegyárusításról). Regisztrálás és bejelentkezés után lehet jegyet online vásárolni az alkalmazáson keresztül; és a bejelentkezett felhasználók többfajta kedvezményben részesülhetnek. Ezenkívül a felhasználók képesek járatot keresni ár, illetve idő alapján és a csatlakozások figyelembevételével keresni.

## Követelménykatalógus

## Városok, állomások, vonatok kezelése

## Menetrend

## Járatkereső *(Lekérdezés)*

## Vásárolt jegyek listázása *(Lekérdezés)*

## Kedvezmények kezelése bejelentkezett felhasználók számára *(Trigger)*

## Kalauzok, jegypénztárosok beosztása

## Beosztásoknál ütköző vagy egymást érő munkarendek ellenőrzése *(Trigger)*

## Szabadságok ellenőrzése *(Trigger)*

## Éves kimutatás készítése *(Lekérdezés)*

## Statisztika készítése a jegyárusításból *(Lekérdezés)*

## Hosszú szabadságok listázása *(Lekérdezés)*

## Alkalmazottak bérének kiszámítása *(Lekérdezés)*

## Törzsvásárlók listázása *(Lekérdezés)*

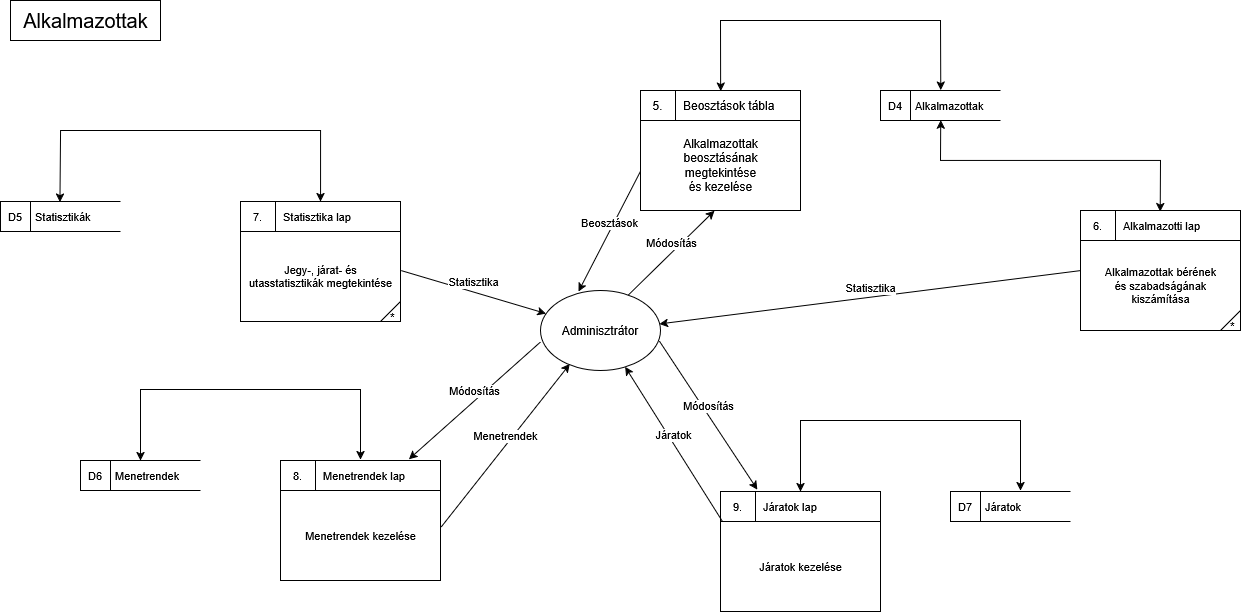
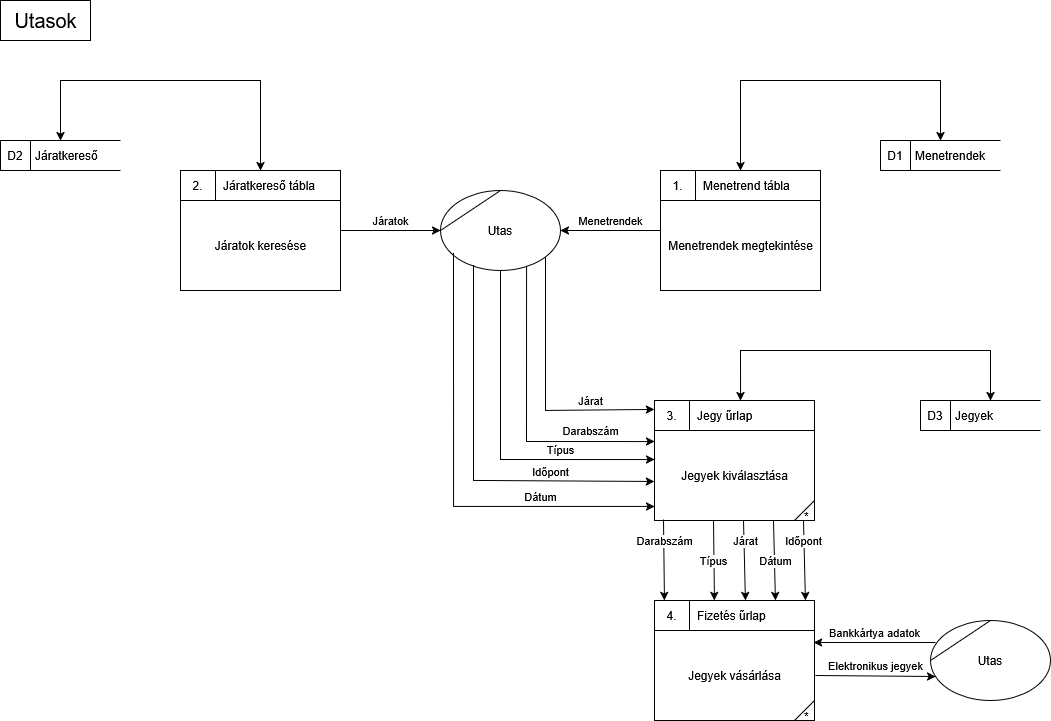
## Felmérése annak, hogy mely vonaton hányan utaztak *(Lekérdezés)*

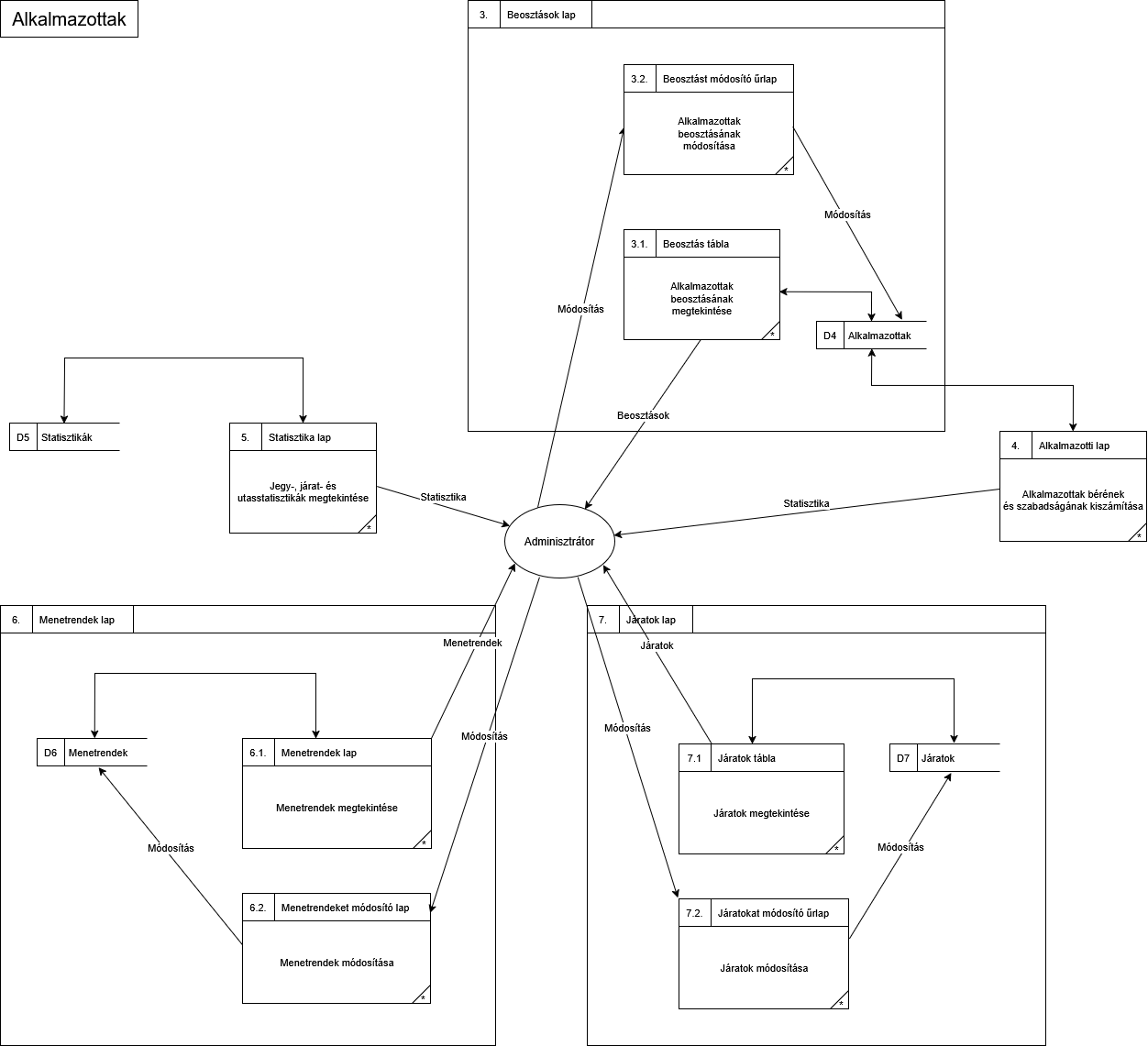
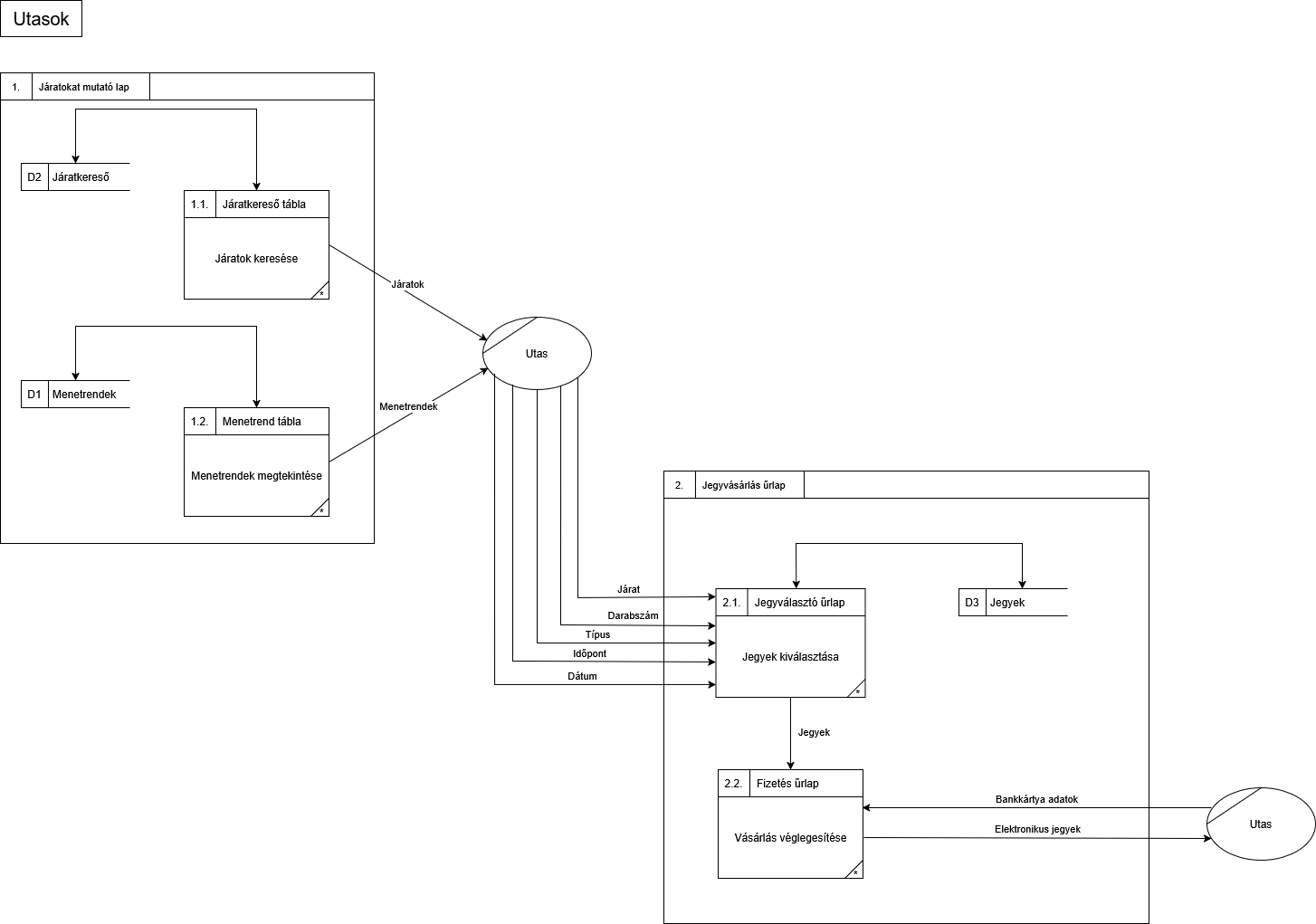
## Új szabadság rögzítése *(Tárolt eljárás)*

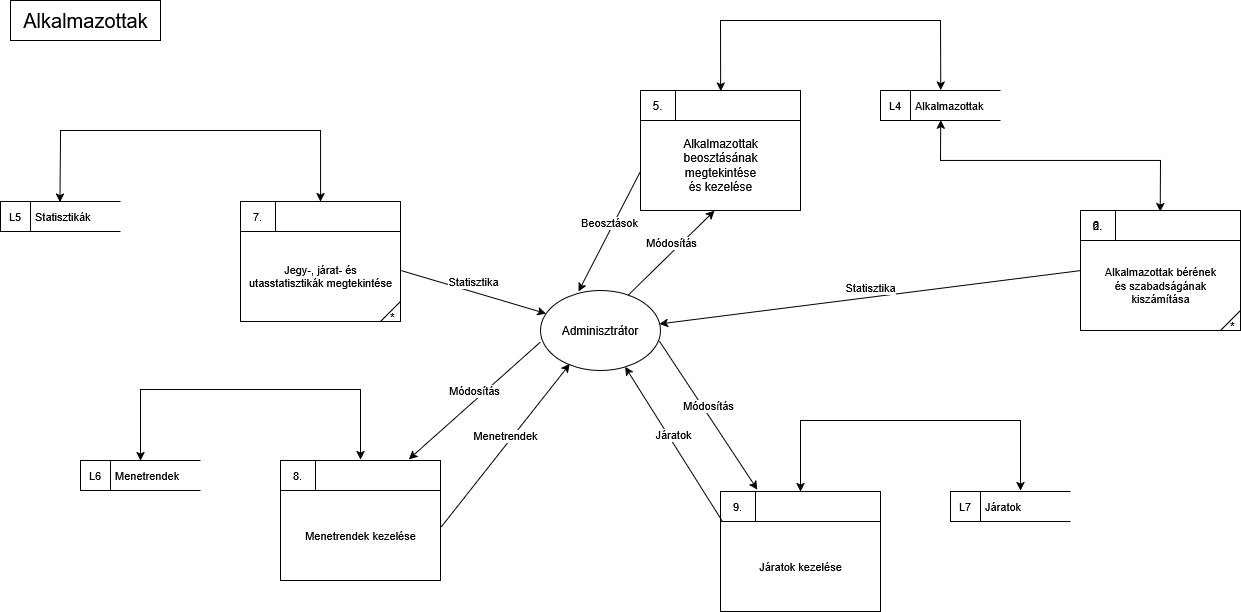
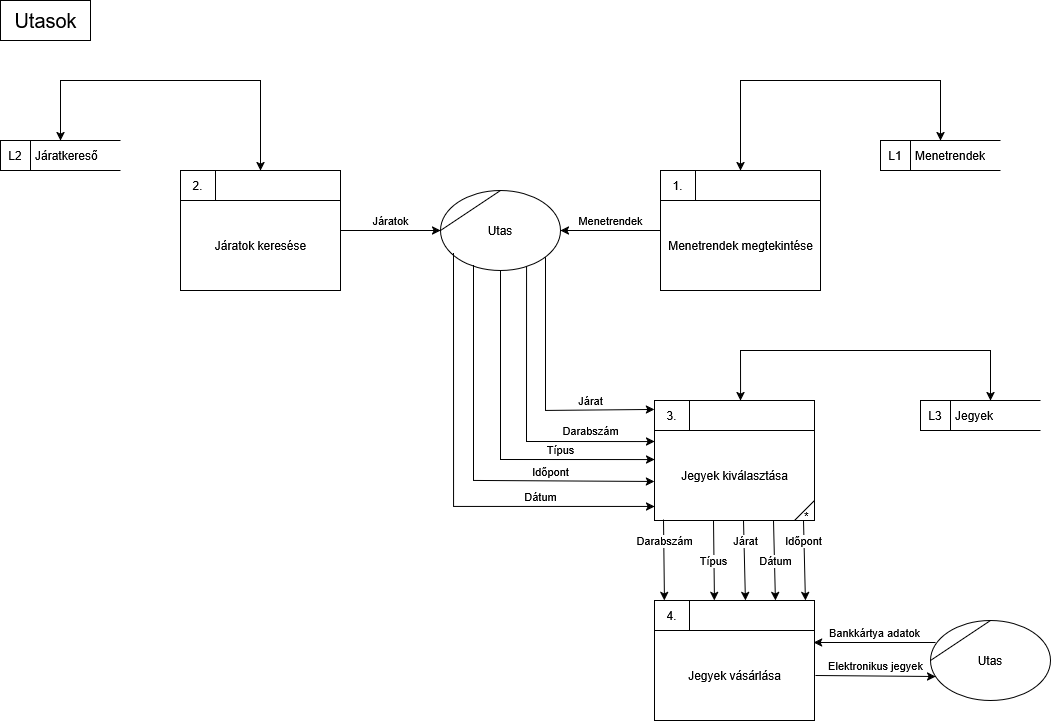
## Felhasználó-kedvezmény statisztika *(Tárolt eljárás)*

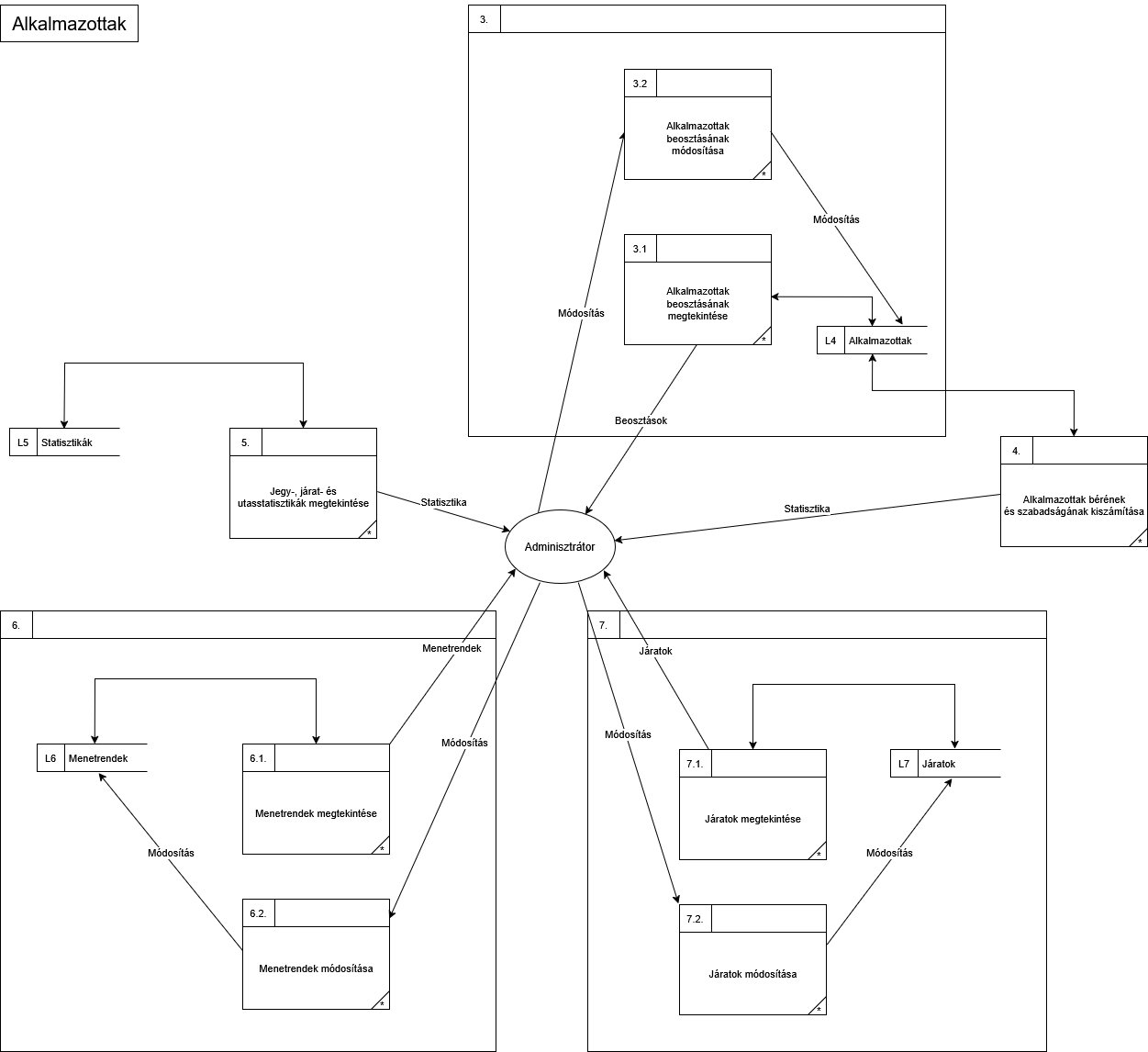
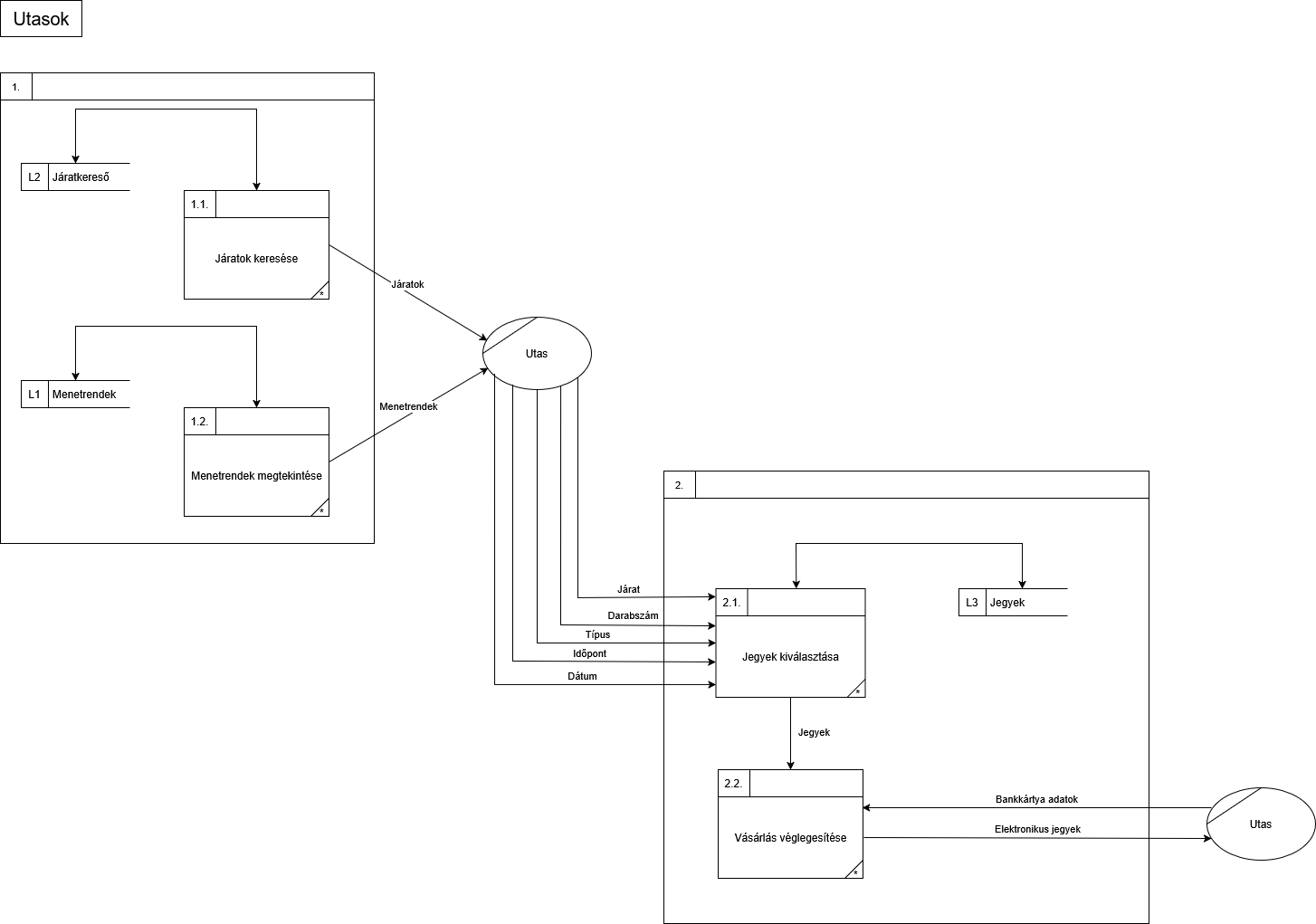
## Online jegyrendelés

## Adatfolyam diagram (DFD):

Fizikai – 1. szint

Fizikai – 2. szint

Logikai – 1. szint

Logikai – 2. szint

## Egyedmodell:

## Egyedmodell.png

## EK-diagram:

## 

## Relációs adatelemzés

Felhasználó (felhasználónév, jelszó, születési idő, alkalmazott, adminisztrátor)

Kedvezmény (k-azonosító, név, kedvezmény százalék)

Jegy (jegy-azonosító, név, ár, felhasználható)

Vásárlás (vásárlás-azonosító, időpont, *felhasználónév*, *k-azonosító*, *jegy-azonosító*, *járat-azonosító*)

Járat (járat-azonosító, indulás, *cs-azonosító*, *vonat-azonosító*)

Vonat (vonat-azonosító, elsőosztályú helyek, másodosztályú helyek)

Csatlakozás (cs-azonosító, időtartam, hossz, *első á-azonosító*. *második á-azonosító*, *járat-azonosító*)

Állomás (á-azonosító, név, város)

Alkalmazott (a-azonosító, név, beosztás, órabér)

Szabadság (sz-azonosító, mettől, meddig, *a-azonosító*)

Munkabeosztás (m-azonosító, milyen nap, kezdet, vég, *a-azonosító*)

1. normálforma:

A relációsémák teljesítik az 1NF feltételeit, mert csak atomi attribútumokat tartalmaznak.

2. normálforma:

A relációsémák teljesítik a 2NF feltételeit, mert csak egy attribútumból álló kulcsokat tartalmaznak.

3. normálforma:

A relációsémák teljesítik a 3NF feltételeit, mert minden másodlagos attribútum közvetlenül függ minden kulcstól.

## Alulról felfelés elemzés:

Vásárlás(vásárlás-azonosító, időpont, felhasználónév*,* születési-idő, jelszó, alkalmazott, adminisztrátor,jegy-név, jegy-ár, jegy-felhasználható, kedvezmény-név, kedvezmény-százalék, indulás, elsőosztályú helyek, másodosztályú helyek, Állomások)

Állomások többértékű attribútum

Alkalmazott(a-azonosító, név, beosztás órabér, Szabadságok, Munkabeosztás)

Szabadságok, Munkabeosztás többértékű attribútumok

1. normálforma:

Vásárlás(vásárlás-azonosító, időpont, felhasználónév*,* születési-idő, jelszó, alkalmazott, adminisztrátor,jegy-név, jegy-ár, jegy-felhasználható, kedvezmény-név, kedvezmény-százalék, indulás, elsőosztályú helyek, másodosztályú helyek, *cs-azonosító*)

Állomás (á-azonosító, név, város)

Csatlakozás (cs-azonosító, időtartam, hossz, *első á-azonosító*. *második á-azonosító*)

Alkalmazott (a-azonosító, név, beosztás, órabér)

Szabadság (sz-azonosító, mettől, meddig, *a-azonosító*)

Munkabeosztás (m-azonosító, milyen nap, kezdet, vég, *a-azonosító*)

2. normálforma:

A relációsémák teljesítik a 2NF feltételeit, mert csak egy attribútumból álló kulcsokat tartalmaznak.

3. normálforma:

Felhasználó (felhasználónév, jelszó, születési idő, alkalmazott, adminisztrátor)

Kedvezmény (k-azonosító, név, kedvezmény százalék)

Jegy (jegy-azonosító, név, ár, felhasználható)

Vásárlás (vásárlás-azonosító, időpont, *felhasználónév*, *k-azonosító*, *jegy-azonosító*, *járat-azonosító*)

Járat (járat-azonosító, indulás, *cs-azonosító*, *vonat-azonosító*)

Vonat (vonat-azonosító, elsőosztályú helyek, másodosztályú helyek)

Csatlakozás (cs-azonosító, időtartam, hossz, *első á-azonosító*. *második á-azonosító*, *járat-azonosító*)

Állomás (á-azonosító, név, város)

Alkalmazott (a-azonosító, név, beosztás, órabér)

Szabadság (sz-azonosító, mettől, meddig, *a-azonosító*)

Munkabeosztás (m-azonosító, milyen nap, kezdet, vég, *a-azonosító*)

## Táblák leírása:

## Felhasználó

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| felhasználónév | VARCHAR2(50) | A felhasználó által választott egyedi felhasználónév |
| jelszó | VARCHAR2(256) | A felhasználó által választott jelszó |
| születési idő | DATE | A felhasználó születési dátuma. A kedvezmények használatához szükséges. |
| alkalmazott | BOOLEAN | A felhasználó alkalmazott –e. A kedvezmények használatához szükséges. |
| adminisztrátor | BOOLEAN | A felhasználó adminisztrátor –e. |

## Kedvezmény

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| k-azonosító | NUMBER | A kedvezmény azonosítója. Ez a kulcs |
| név | VARCHAR2(100) | A kedvezmény neve |
| kedvezmény százalék | NUMBER | A teljes ár hány százalékát kell kifizetni, ha használjuk a kedvezményt. |

## Jegy

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| jegy-azonosító | NUMBER | A jegy azonosítója. Ez a kulcs |
| név | VARCHAR2(100) | A jegy neve |
| ár | NUMBER | A jegy ára. |
| felhasználható | NUMBER | Hány napig felhasználható a jegy |

## Vásárlás

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| vásárlás-azonosító | NUMBER | A vásárlás azonosítója. Ez a kulcs |
| időpont | TIMESTAMP | A jegy érvényességének kezdete |
| felhasználónév | VARCHAR2(50) | A vásárló felhasználóneve |
| k-azonosító | NUMBER | A felhasznált kedvezmény azonosítója |
| jegy-azonosító | NUMBER | A megvásárolt jegytípus azonosítója (menetjegy, napijegy stb.) |
| járat-azonosító | NUMBER | A járat, amelyre szól a jegy (időszakos jegy esetén Null) |

## Járat

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| járat-azonosító | NUMBER | A járat azonosítója. Ez a kulcs |
| indulás | TIMESTAMP | A járat indulásának időpontja |
| cs-azonosító | NUMBER | Az első csatlakozás azonosítója, amely tartalmazza az első és a második állomást |
| vonat-azonosító | NUMBER | Az utasok ezen a vonaton fognak utazni, ha a járatra vesznek jegyet |

## Vonat

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| vonat-azonosító | NUMBER | A vonat azonosítója. Ez a kulcs |
| elsőosztályú helyek | NUMBER | Az elsőosztályú férőhelyek száma |
| másodosztályú helyek | NUMBER | Az másodosztályú férőhelyek száma |

## Csatlakozás

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| cs-azonosító | NUMBER | A csatlakozás azonosítója. Ez a kulcs |
| időtartam | NUMBER | A két állomás közötti útszakasz időtartama percben |
| hossz | NUMBER | A két állomás közötti útszakasz hossza kilóméterben |
| első á-azonosító | NUMBER | Az első állomás azonosítója |
| második á-azonosító | NUMBER | A második állomás azonosítója |
| járat-azonosító | NUMBER | A csatlakozás ennek a járatnak a része |

## Állomás

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| á-azonosító | NUMBER | Az állomás azonosítója. Ez a kulcs |
| név | VARCHAR2(100) | Az állomás neve |
| város | VARCHAR2(100) | A város neve, ahol az állomás van |

## Alkalmazott

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| a-azonosító | NUMBER | Az alkalmazott azonosítója. Ez a kulcs |
| név | VARCHAR2(100) | Az alkalmazott neve |
| beosztás | VARCHAR2(50) | Az alkalmazott beosztása |
| órabér | NUMBER | Az alkalmazott órabére |

## Szabadság

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| sz-azonosító | NUMBER | Az szabadság azonosítója. Ez a kulcs |
| mettől | DATE | Az alkalmazott ettől kezdve van szabadságon |
| meddig | DATE | Az alkalmazott eddig van szabadságon |
| a-azonosító | NUMBER | Melyik alkalmazottra vonatkozik a szabadság |

## Munkabeosztás

|  |  |  |
| --- | --- | --- |
| **Név** | **Típus** | **Leírás** |
| m-azonosító | NUMBER | Az munkabeosztás azonosítója. Ez a kulcs |
| milyen nap | DATE | A hét hányadik napjára vonatkozik a beosztás |
| kezdet | NUMBER | A munkavégzés kezdete 00:00-tól számítva percekben |
| vég | NUMBER | A munkavégzés vége 00:00-tól számítva percekben |
| a-azonosító | NUMBER | Melyik alkalmazottra vonatkozik a beosztás |

## Szerep-funkció mátrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Menetrendek megtekintése** | **Menetrendek módosítása** | **Járatok megtekintése** | **Járatok módosítása** | **Jegyárúsítási stat. megtekintése** | **Alkalmazottak beoszt. megtekintése** | **Alkalmazottak beoszt. módosítás** | **Alkalmazottak bér és szabadság kiszámítása** | **Járat kiválasztása** | **Ülések kiválasztás** | **Jegyek megvásárlása** | **Jegyek kiválasztása** | **Járatok keresése ár és idő szerint** |
| Utas | x |  | x |  |  |  |  |  | x | x | x | x | x |
| Admin | x | x | x | x | x | x | x | x |  |  |  |  |  |
| Kalauz | x |  | x |  | x | x |  |  |  |  |  |  |  |
| Pénztáros | x |  | x |  | x | x |  |  |  |  |  |  | x |

## Egyed-esemény mátrix:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Események** | **Menetrendek megtekintése** | **Menetrendek módosítása** | **Járatok megtekintése** | **Járatok módosítása** | **Jegyárúsítási stat. megtekintése** | **Alkalmazottak beoszt. megtekintése** | **Alkalmazottak beoszt. módosítás** | **Alkalmazottak bér és szabadság kiszámítása** | **Járat kiválasztása** | **Ülések kiválasztás** | **Jegyek megvásárlása** | **Jegyek kiválasztása** | **Járatok keresése ár és idő szerint** |
| **Egyedek** |  |  |  |  |  |  |  |
| Alkalmazott | O | L,M,T | O | L,M,T |  | O | O | O |  |  |  |  |  |
| Jegy |  |  |  |  | O |  |  |  |  |  | L,O,T | O | O |
| Vásárlás |  |  |  |  | O |  |  |  |  |  | L,T | O |  |
| Munkabeosztás |  |  |  |  |  | O | O,L,M,T | O |  |  |  |  |  |
| Szabadság |  |  |  |  |  | O | M | O |  |  |  |  |  |
| Felhasználó | O | L,M,T | O | L,M,T |  | O |  |  |  | O | O | O |  |
| Vonat |  | L,M,T |  | L,M,T |  |  |  |  | O | O,M |  |  |  |
| Kedvezmény |  |  |  |  | O |  |  |  |  |  | O,L | O | O |
| Járat | O | L,M,T | O | L,M,T | O |  |  |  | O | O |  | O | O |
| Csatlakozás | O | L,M,T | O | L,M,T |  |  |  |  | O |  |  |  | O |
| Állomás | O | L,M,T | O | L,M,T |  |  |  |  | O |  |  |  | O |

## Funkciómeghatározás (amit lehet az oldalon csinálni)

**Vásárlói/Utas funkciók**

|  |  |
| --- | --- |
| Funkció azonosító: | 1 |
| Funkciónév: | Menetrendek megtekintése |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Utasok, Utas |
| Funkció leírása: | Menetrend megmutatja, hogy melyik vonat mikor megy, honnan hova mit érintve. |
| Hibakezelés: | A járat megszűnt, nem létezik. |
| AFD eljárások: | 1 |
| Köv. katalógusra hivatkozás: | Menetrend |
| Tömegszerűség: | kb. napi több 1000 |
| Lekérdezések: | Járatok, állomások, vonatok, idő |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 2 |
| Funkciónév: | Járatok keresése ár és idő szerint |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Utasok, Utas |
| Funkció leírása: | Járatok keresése ár és idő szerint, mikor indul, és mennyibe kerülnek a különböző jegyek. |
| Hibakezelés: | A járat megszűnt, nem létezik. Nem lehet negatív összeg alapján keresni. |
| AFD eljárások: | 2 |
| Köv. katalógusra hivatkozás: | Vonatkereső idő és ár szerint |
| Tömegszerűség: | kb. napi több 100 |
| Lekérdezések: | Járatok, állomások, vonatok, idő, ár |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 3 |
| Funkciónév: | Jegyek kiválasztása |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Utasok, Utas |
| Funkció leírása: | Az utas elektronikusan jegyet választ, mennyit, milyet, melyik járatra, mikor, milyen osztályra stb. |
| Hibakezelés: | Nem adhat meg régebbi dátumot, mint a mai. |
| AFD eljárások: | 3 |
| Köv. katalógusra hivatkozás: | Menetjegy, pótjegy vásárlása |
| Tömegszerűség: | kb. napi több 1000 |
| Lekérdezések: | Járatok, állomások, vonatok, idő, ár |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 4 |
| Funkciónév: | Jegyek vásárlása |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Utasok, Utas |
| Funkció leírása: | Az utas elektronikusan jegyet vesz, a bankkártya adatait megadva. |
| Hibakezelés: | Nincs megadva bankkártya, nincs rajta elég pénz. |
| AFD eljárások: | 4 |
| Köv. katalógusra hivatkozás: | Menetjegy, pótjegy vásárlása |
| Tömegszerűség: | kb. napi több 1000 |
| Lekérdezések: | Vásárló bankkártya adatainak lekérdezése |
| Válaszidő/tűrés: | 3 sec / 10 sec |

**Alkalmazotti funkciók**

|  |  |
| --- | --- |
| Funkció azonosító: | 1 |
| Funkciónév: | Alkalmazottak beosztásának megtekintése és kezelése |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Alkalmazott, Adminisztrátor |
| Funkció leírása: | Alkalmazottak beosztásának megtekintése és kezelése |
| Hibakezelés: | A beosztásban ütközések lehetnek |
| AFD eljárások: | 1 |
| Köv. katalógusra hivatkozás: | Kalauzok, jegypénztárosok beosztásánál az ütköző vagy egymást érő munkarendek ellenőrzése. Kalauzok, jegypénztárosok beosztása |
| Tömegszerűség: | kb. napi 3-4 |
| Lekérdezések: | Alkalmazottak, beosztás, szabadságok |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 2 |
| Funkciónév: | Alkalmazottak bérének és szabadságának kiszámítása |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Alkalmazott, Adminisztrátor |
| Funkció leírása: | Alkalmazottak bérének és szabadságának kiszámítása, ledolgozott órák, munkabér, táppénz és elhasznált valamint megmaradt szabadságok függvényében. |
| Hibakezelés: | Nem lehet negatív a bér és a szabadságok száma. |
| AFD eljárások: | 2 |
| Köv. katalógusra hivatkozás: | Alkalmazotti bérek, szabadságok kiszámítása |
| Tömegszerűség: | kb. heti 1-2 |
| Lekérdezések: | Alkalmazottak, beosztás, szabadságok, bér/fizetés |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 3 |
| Funkciónév: | Jegyárusítási statisztikák megtekintése |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Alkalmazott, Adminisztrátor |
| Funkció leírása: | Jegyárusítási statisztikák megtekintése, milyen jegyből, mennyit, mikor. |
| Hibakezelés: |  |
| AFD eljárások: | 3 |
| Köv. katalógusra hivatkozás: | Statisztika készítése a jegyárusításból |
| Tömegszerűség: | kb. Évi 3-4 |
| Lekérdezések: | Jegy, járat, vonat |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 4 |
| Funkciónév: | Menetrendek kezelése |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Alkalmazott, Adminisztrátor |
| Funkció leírása: | Menetrendek kezelése: melyik járat mikor, milyen gyakran jár, hol áll meg stb. |
| Hibakezelés: | A járat/város/állomás/útvonal megszűnt/nem létezik |
| AFD eljárások: | 4 |
| Köv. katalógusra hivatkozás: | Menetrend, városok, állomások, vonatok kezelése |
| Tömegszerűség: | kb. heti 3-4 |
| Lekérdezések: | Menetrendek, járat, város, állomás |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 5 |
| Funkciónév: | Járatok kezelése |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Alkalmazott, Adminisztrátor |
| Funkció leírása: | Járatok lekérdezése, melyik járat mikor, milyen gyakran jár, hol áll meg stb. |
| Hibakezelés: | A járat megszűnt/nem létezik |
| AFD eljárások: | 5 |
| Köv. katalógusra hivatkozás: | Városok, állomások, vonatok kezelése |
| Tömegszerűség: | kb. heti 3-4 |
| Lekérdezések: | Menetrendek, járat, város, állomás |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 6 |
| Funkciónév: | Járatok használtsági statisztikája |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Adminisztrátor |
| Funkció leírása: | Lekérdezi, hogy az adott járatot hányan használták a megadott idő intervallumban |
| Hibakezelés: | Hiányos járat adatok, törölt járatnak megmaradt adatai |
| AFD eljárások: | 3 |
| Köv. katalógusra hivatkozás: | Felmérése annak, hogy mely vonaton hányan utaztak |
| Tömegszerűség: | kb. heti 2-3 |
| Lekérdezések: | Járat, Vonat, Vásárlás |
| Válaszidő/tűrés: | 3 sec / 10 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 7 |
| Funkciónév: | Éves kimutatás készítése |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Adminisztrátor |
| Funkció leírása: | Lekérdezi a költségeket, a nyereségeket, és ebből kiszámítja a nyereséget/veszteséget a jelen évre |
| Hibakezelés: | Hiányos adatok |
| AFD eljárások: | 3 |
| Köv. katalógusra hivatkozás: | Éves kimutatás készítése |
| Tömegszerűség: | kb. hónapi 1 |
| Lekérdezések: | Munkabeosztás, Alkalmazott, Vásárlás, Jegy, Kedvezmény |
| Válaszidő/tűrés: | 2 sec / 5 sec |

|  |  |
| --- | --- |
| Funkció azonosító: | 8 |
| Funkciónév: | Törzsvásárlók listázása |
| Típus: | online,lekérdezés,felhasználói |
| Felhasználói szerepkörök: | Adminisztrátor |
| Funkció leírása: | Lekérdezi azokat a vásárlókat, akik egy adott jegyszám felett vásároltak |
| Hibakezelés: | Hiányos adatok |
| AFD eljárások: | 3 |
| Köv. katalógusra hivatkozás: | Törzsvásárlók listázása |
| Tömegszerűség: | kb. heti 1 |
| Lekérdezések: | Felhasználó, Vásárlás |
| Válaszidő/tűrés: | 2 sec / 5 sec |

## Képernyőtervek (Pótolható)

## Menütervek (Pótolható)

## Összetett lekérdezések

## Járatkereső csatlakozások figyelembevételével

## app.py : 1153. sor

## Utas 2. funkció

## Utas fizikai 2.szint 1. űrlap

@app.route("/api\_jaratkereso")

def api\_jaratkereso():

    connection, cursor = get\_db()

    destination\_loc = request.args.get('jaratkereso\_hova', type=str)

    result = []

    for row in cursor.execute(

        f"SELECT Jarat.jarat\_azonosito, nev FROM Jarat, Allomas, Csatlakozas WHERE Jarat.jarat\_azonosito = Csatlakozas.jarat\_azonosito AND a\_azonosito IN (SELECT a\_azonosito FROM Allomas, Csatlakozas WHERE a\_azonosito = masodik\_a\_azonosito AND nev = '{destination\_loc}')"

    ):

        result.append({

            "jarat\_azonosito": row[0],

            "nev": row[1]

        })

    return jsonify(result)

## Vásárolt jegyek listázása

## app.py: 1299. sor

## Alkalmazott 3. funkció

## Alkalmazott fizikai 2. szint 5. lap

@app.route("/api\_jegyvasarlas")

def api\_jegyvasarlas():

    connection, cursor = get\_db()

    kedvezmeny = request.args.get('jvasar\_kedv', type=int)

    jegy = request.args.get('jvasar\_jegy', type=int)

    jarat = request.args.get('jvasar\_jarat', type=int)

    result = []

    for row in cursor.execute(

        f"SELECT vasarlas\_azonosito, felhasznalonev, idopont FROM Vasarlas, Kedvezmeny, Jegy, Jarat WHERE Vasarlas.k\_azonosito = Kedvezmeny.k\_azonosito AND Kedvezmeny.k\_azonosito = {kedvezmeny} AND Vasarlas.jegy\_azonosito = Jegy.jegy\_azonosito AND Jegy.jegy\_azonosito = {jegy} AND Vasarlas.jarat\_azonosito = Jarat.jarat\_azonosito AND Jarat.jarat\_azonosito = {jarat} GROUP BY felhasznalonev, idopont, vasarlas\_azonosito Having COUNT(felhasznalonev) > 0"

    ):

        result.append({

            "vasarlas\_azonosito": row[0],

            "felhasznalonev": row[1],

            "idopont": row[2]

        })

    return jsonify(result)

## Éves kimutatás készítése

## app.py : 1188. sor

## Alkalmazott 7. funkció

## Alkalmazott fizikai 2.szint 5. lap

@app.route("/api\_eveskimutatas")

def api\_eveskimutatas():

    connection, cursor = get\_db()

    cursor.execute("""

            SELECT

                NVL((

                    SELECT SUM((mb.veg - mb.kezdet) \* a.oraber)

                    FROM Munkabeosztas mb

                    JOIN Alkalmazott a ON mb.a\_azonosito = a.a\_azonosito

                    WHERE EXTRACT(YEAR FROM mb.milyen\_nap) = TO\_NUMBER(TO\_CHAR(SYSDATE, 'YYYY'))

                ), 0) AS dolgozoi\_koltseg,

                NVL((

                    SELECT SUM(j.ar \* (1 - NVL(k.kedvezmeny\_szazalek, 0)/100))

                    FROM Vasarlas v

                    JOIN Jegy j ON v.jegy\_azonosito = j.jegy\_azonosito

                    LEFT JOIN Kedvezmeny k ON v.k\_azonosito = k.k\_azonosito

                    WHERE EXTRACT(YEAR FROM v.idopont) = TO\_NUMBER(TO\_CHAR(SYSDATE, 'YYYY'))

                ), 0) AS jegy\_bevetel,

                (

                    NVL((

                        SELECT SUM(j.ar \* (1 - NVL(k.kedvezmeny\_szazalek, 0)/100))

                        FROM Vasarlas v

                        JOIN Jegy j ON v.jegy\_azonosito = j.jegy\_azonosito

                        LEFT JOIN Kedvezmeny k ON v.k\_azonosito = k.k\_azonosito

                        WHERE EXTRACT(YEAR FROM v.idopont) = TO\_NUMBER(TO\_CHAR(SYSDATE, 'YYYY'))

                    ), 0)

                    -

                    NVL((

                        SELECT SUM((mb.veg - mb.kezdet) \* a.oraber)

                        FROM Munkabeosztas mb

                        JOIN Alkalmazott a ON mb.a\_azonosito = a.a\_azonosito

                        WHERE EXTRACT(YEAR FROM mb.milyen\_nap) = TO\_NUMBER(TO\_CHAR(SYSDATE, 'YYYY'))

                    ), 0)

                ) AS nyereseg

            FROM dual

        """)

    row = cursor.fetchone()

    result = {

        "dolgozoi\_koltseg": row[0],

        "jegy\_bevetel": row[1],

        "nyereseg": row[2]

    }

    return jsonify(result)

## Statisztika készítése a jegyárusításból

## app.py: 1238. sor

## Alkalmazott 3. funkció

## Alkalmazott fizikai 2.szint 5. lap

@app.route("/api\_arustat")

def api\_arustat():

    connection, cursor = get\_db()

    jegy = request.args.get('stat\_jegy', type=int)

    year = request.args.get('stat\_ev', type=int)

    result = []

    for row in cursor.execute(

        f"SELECT DISTINCT vasarlas\_azonosito, felhasznalonev, idopont, ar, kedvezmeny\_szazalek FROM Vasarlas, Jegy, Kedvezmeny WHERE Vasarlas.jegy\_azonosito = Jegy.jegy\_azonosito AND Vasarlas.jegy\_azonosito = {jegy} AND EXTRACT(year FROM idopont) = {year} GROUP BY vasarlas\_azonosito, felhasznalonev, idopont, ar, kedvezmeny\_szazalek HAVING COUNT(vasarlas\_azonosito) > 0"

    ):

        result.append({

            "vasarlas\_azonosito": row[0],

            "felhasznalonev": row[1],

            "idopont": row[2],

            "eredeti ar": row[3],

            "kedvezmenyes ar": row[3] \* row[4] / 100

        })

    return jsonify(result)

## Hosszú szabadságok listázása (5 napnál hosszabb)

## app.py: 1259. sor

## Alkalmazott 2. funkció

## Alkalmazott fizikai 2.szint 5. lap

@app.route("/api\_hosszuszab")

def api\_hosszuszab():

    connection, cursor = get\_db()

    year = request.args.get('nemd\_ev', type=int)

    month = request.args.get('nemd\_honap', type=int)

    result = []

    for row in cursor.execute(

        f"SELECT DISTINCT Alkalmazott.a\_azonosito, nev, beosztas FROM Alkalmazott, Szabadsag WHERE EXTRACT(year FROM mettol) = {year} AND EXTRACT(month FROM mettol) = {month} AND Alkalmazott.a\_azonosito IN (SELECT Alkalmazott.a\_azonosito FROM Alkalmazott, Szabadsag WHERE Alkalmazott.a\_azonosito = Szabadsag.a\_azonosito AND (EXTRACT(day FROM meddig) - EXTRACT(day FROM mettol)) > 5)"

    ):

        result.append({

            "a\_azonosito": row[0],

            "nev": row[1],

            "beosztas": row[2]

        })

    return jsonify(result)

## Alkalmazottak bérének kiszámítása

## app.py : 1278.sor

## Alkalmazott 2. funkció

## Alkalmazott fizikai 2. szint 4. lap

@app.route("/api\_berszam")

def api\_berszam():

    connection, cursor = get\_db()

    year = request.args.get('ber\_ev', type=int)

    month = request.args.get('ber\_honap', type=int)

    result = []

    for row in cursor.execute(

        f'SELECT Alkalmazott.a\_azonosito, nev, beosztas, kezdet, veg, oraber FROM Alkalmazott, Munkabeosztas WHERE Alkalmazott.a\_azonosito = Munkabeosztas.a\_azonosito AND EXTRACT(year FROM milyen\_nap) = {year} AND EXTRACT(month FROM milyen\_nap) = {month} GROUP BY Alkalmazott.a\_azonosito,  nev, beosztas, kezdet, veg, oraber HAVING COUNT(Alkalmazott.a\_azonosito) > 0'

    ):

        result.append({

            "a\_azonosito": row[0],

            "nev": row[1],

            "beosztas": row[2],

            "munkaido (ora)": (row[4] - row[3]) / 60,

            "ber": ((row[4] - row[3]) / 60) \* row[5]

        })

    return jsonify(result)

## Törzsvásárlók listázása

## app.py: 1321. sor

## Alkalmazott 8. funckió

## Alkalmazott fizikai 2. szint 5. lap

@app.route("/api\_torzsvasarlo")

def api\_torzsvasarlo():

    connection, cursor = get\_db()

    result = []

    for row in cursor.execute(

         f"SELECT Felhasznalo.felhasznalonev, szuletesi\_ido FROM Felhasznalo, Vasarlas WHERE Felhasznalo.felhasznalonev = Vasarlas.felhasznalonev AND alkalmazott = 0 AND administrator = 0 GROUP BY Felhasznalo.felhasznalonev, szuletesi\_ido HAVING COUNT(vasarlas\_azonosito) > 20"

    ):

        result.append({

            "felhasznalonev": row[0],

            "szuletesi\_ido": row[1]

        })

    return jsonify(result)

## Felmérése annak, hogy mely vonaton hányan utaztak

## app.py : 1170.sor

## Alkalmazott 6. funkció

## Alkalmazott fizikai 2. szint 5. lap

@app.route("/api\_utasszam")

def api\_utasszam():

    connection, cursor = get\_db()

    year = request.args.get('utasszam\_ev', type=int)

    result = []

    for row in cursor.execute(

        f'SELECT Jarat.jarat\_azonosito, elso\_osztalyu\_helyek, masod\_osztalyu\_helyek, SUM(vasarlas\_azonosito) FROM Jarat, Vonat, Vasarlas WHERE Jarat.jarat\_azonosito = Vasarlas.jarat\_azonosito AND EXTRACT(year FROM idopont) = {year} GROUP BY Jarat.jarat\_azonosito, elso\_osztalyu\_helyek, masod\_osztalyu\_helyek HAVING COUNT(Jarat.jarat\_azonosito) > 0'

    ):

        result.append({

            "jarat\_azonosito": row[0],

            "utasszam": row[3],

            "max\_hely": row[1] + row[2]

        })

    return jsonify(result)

**Triggerek**

## Kedvezmény kezelő

## tabla\_letrehozo.sql : 136.sor

CREATE OR REPLACE TRIGGER kedvezmeny\_trigger

BEFORE INSERT

ON Vasarlas

FOR EACH ROW

DECLARE

    kNev VARCHAR2(100);

    fEv NUMBER;

    fAlk NUMBER;

BEGIN

    SELECT nev INTO kNev FROM Kedvezmeny WHERE k\_azonosito = :NEW.k\_azonosito;

    SELECT floor(months\_between(SYSDATE, CAST(szuletesi\_ido AS DATE)) /12) INTO fEv FROM Felhasznalo WHERE felhasznalonev = :NEW.felhasznalonev;

    SELECT alkalmazott INTO fAlk FROM Felhasznalo WHERE felhasznalonev = :NEW.felhasznalonev;

    IF kNev = 'Diák' AND fEv > 18 THEN

        :NEW.k\_azonosito := NULL;

    ELSIF kNev = 'Nyugdíjas' AND fEv < 65 THEN

        :NEW.k\_azonosito := NULL;

    ELSIF kNev = 'MÁK-ONYF' AND fAlk = 0 THEN

        :NEW.k\_azonosito := NULL;

    END IF;

END;

/

## Szabadság ellenőrző

## tabla\_letrehozo.sql : 386. sor

CREATE OR REPLACE TRIGGER szabadsag\_duplikacio\_ellenorzes\_trigger

BEFORE INSERT OR UPDATE ON Szabadsag

FOR EACH ROW

DECLARE

    van\_egyezes NUMBER;

BEGIN

    SELECT COUNT(\*) INTO van\_egyezes

    FROM Szabadsag

    WHERE a\_azonosito = :NEW.a\_azonosito

      AND (

            (:NEW.mettol BETWEEN mettol AND meddig) OR

            (:NEW.meddig BETWEEN mettol AND meddig) OR

            (mettol BETWEEN :NEW.mettol AND :NEW.meddig) OR

            (meddig BETWEEN :NEW.mettol AND :NEW.meddig)

          )

      AND sz\_azonosito != :NEW.sz\_azonosito; -- Saját rekord kivétele UPDATE esetén

    IF van\_egyezes > 0 THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'Ez az alkalmazott már rendelkezik szabadsággal ezen a napon vagy átfedésben.');

    END IF;

END;

/

Beosztás ellenőrző

## tabla\_letrehozo.sql : 163. sor

CREATE OR REPLACE TRIGGER beosztas\_ellenorzes

BEFORE INSERT OR UPDATE OF kezdet, veg ON munkabeosztas

FOR EACH ROW

DECLARE

    v\_szamlalo NUMBER;

BEGIN

    SELECT COUNT(\*)

    INTO v\_szamlalo

    FROM munkabeosztas

    WHERE a\_azonosito = :NEW.a\_azonosito

      AND milyen\_nap = :NEW.milyen\_nap

      AND m\_azonosito != :NEW.m\_azonosito

      AND (

          (:NEW.kezdet BETWEEN kezdet AND veg)

          OR (:NEW.veg BETWEEN kezdet AND veg)

          OR (kezdet BETWEEN :NEW.kezdet AND :NEW.veg)

      )

      AND (:NEW.rowid IS NULL OR ROWID != :NEW.ROWID); -- hogy UPDATE-nél ne önmagát nézze

    IF v\_szamlalo > 0 THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'Ütköző vagy átfedő beosztás létezik erre az alkalmazottra.');

    END IF;

END;

/

## Tárolt eljárások/függvények

## Új szabadság rögzítése

## tabla\_letrehozo.sql : 196. sor

CREATE OR REPLACE PROCEDURE Uj\_Szabadsag\_Rogzitese (

    p\_a\_azonosito IN NUMBER,

    p\_mettol IN DATE,

    p\_meddig IN DATE

) AS

    v\_count NUMBER;

BEGIN

    -- Ellenőrizzük, van-e átfedés

    SELECT COUNT(\*) INTO v\_count

    FROM Szabadsag

    WHERE a\_azonosito = p\_a\_azonosito

      AND (

           (p\_mettol BETWEEN mettol AND meddig)

        OR (p\_meddig BETWEEN mettol AND meddig)

        OR (p\_mettol <= mettol AND p\_meddig >= meddig)

      );

    IF v\_count > 0 THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'Átfedő szabadság már létezik.');

    ELSE

        INSERT INTO Szabadsag (mettol, meddig, a\_azonosito)

        VALUES (p\_mettol, p\_meddig, p\_a\_azonosito);

    END IF;

END;

/

## Felhasználó-kedvezmény statisztika

## tabla\_letrehozo.sql: 566. sor

CREATE OR REPLACE PROCEDURE Felhasznalo\_Kedvezmeny\_Statisztika (

    p\_eredmeny OUT tipusok\_pkg.stat\_cursor

) AS

BEGIN

    OPEN p\_eredmeny FOR

        SELECT

            f.felhasznalonev,

            k.nev AS kedvezmeny\_nev,

            COUNT(\*) AS hasznalatok\_szama

        FROM

            Vasarlas v

        JOIN Felhasznalo f ON v.felhasznalonev = f.felhasznalonev

        JOIN Kedvezmeny k ON v.k\_azonosito = k.k\_azonosito

        GROUP BY

            f.felhasznalonev, k.nev

        ORDER BY

            f.felhasznalonev, hasznalatok\_szama DESC;

END;

/

## Az alkalmazás telepítése

## Az alkalmazás webes felületen keresztül elérhető.

## Szükséges szoftver:

## Python + a requirements.txt-ben szereplő module-ok

## SQL Developer

## Parancssor/terminál/Powershell

## Telepítési utasítások:

1. Töltsd le a repót
2. Telepítsd a \*requirements.txt\*-ből a module-okat (pip install -r requirements.txt)
3. Futtasd le a \*tabla\_letrehozo.sql\* szkriptet az SQL Developer alkalmazásban
4. Nyiss SSH csatlakozást az orania2 szerverre
5. Nyisd meg a parancssorban/terminálban a src mappát
6. Futtasd a \*flask run\* parancsot
7. Add meg az Oracle adatbázis adataidat

További információk a README.MD fájlban.