

# Adatbázis alapú rendszerek

2022-2023/2  
IB152L-1

**Könyvesbolt**

**Készítette:**  
Bakk Ábel  
Ocztos Károly Levente

## Munka felosztása

A csapatot két ember alkotja, **Bakk Ábel** és **Ocztos Károly Levente**.

1. Mérföldkő: Dokumentáció és adatbázisterv bemutatása
  - a. Feladat szöveges leírása, követelménykatalógus – OK
  - b. Adatfolyam diagram – OK
  - c. Egyedmodell, EK-diagram – BÁ
  - d. Relációs adatelemzés – BÁ
  - e. Szerep-funkció mátrix – OK
  - f. Egyed-esemény mátrix – BÁ
  - g. Funkciómeghatározás – OK
  - h. Képernyőtervezek – BÁ
2. Mérföldkő: Adatbázist létrehozó szkriptek beadása
  - a. Táblák létrehozása, táblák közötti kapcsolatok megvalósítása – BÁ
  - b. Feltöltés rekordokkal (táblánként min. 30, kivéve ahol nem indokolt → cserében máshol több) – OK
3. Mérföldkő: Adatlekérés bemutatása grafikus felhasználói felületen
  - a. Entity-k, Repository-k létrehozása – BÁ
  - b. GUI templatek:
    - i. Navigációs menü létrehozása (fragment) - OK
    - ii. Footer létrehozása (fragment) - OK
    - iii. Főoldal létrehozása (képernyőtervezek alapján) - OK
    - iv. Újdonság oldal létrehozása - BÁ
    - v. Sikerlista oldal létrehozása - BÁ
    - vi. Áruházak oldal létrehozása – OK
    - vii. Kívánságlista oldal létrehozása - OK
    - viii. Kosár létrehozása - OK
    - ix. Felhasználói profil létrehozása - OK
    - x. Figyelmeztetések listázása - BÁ
    - xi. Adminisztrációs panel (csak listázás) – BÁ
  - c. Üzleti logika (Service- és Controller layer)
    - i. Felhasználói regisztráció - OK
    - ii. Könyvek listázása (főoldalra, filterezés műfajuk szerint, legnépszerűbb könyvek műfajonként, keresés címre, szerzőre, műfajra, találatok számának megjelenítése, olcsó könyvek listázása, ajánlások) - BÁ
    - iii. Áruházak listázása, adott könyv mely áruházban kapható, mely könyvek kaphatóak egy áruházban, áruházi készlet listázása - OK
    - iv. Adminisztrátornak listázni a felhasználókat - OK

- v. Kívánságlista listázása - OK
4. Mérködő: Adatfelvitel bemutatása grafikus felhasználói felületen
- a. GUI:
    - i. Adminisztrációs panel CRUD műveletekhez – BÁ
    - ii. Kosár továbbfejlesztése – BÁ
    - iii. Kívánságlista továbbfejlesztése - OK
    - iv. Felhasználói profil továbbfejlesztése (pl. adatok módosítása) - OK
    - v. A jelenlegi view elemekhez hozzáadni az eljárások által produkált extra információkat. - közös
  - b. Üzleti logika továbbfejlesztése (alapvetően CRUD műveletek megvalósítása)
    - i. Könyvek felvétele és kezelése – BÁ
    - ii. Műfajok felvétele és kezelése – BÁ
    - iii. Írók felvétele és kezelése – OK
    - iv. Áruházak felvétele és kezelése – OK
    - v. Ügyfelek felvétele és kezelése – OK
    - vi. Kívánságlista felvétele és kezelése – BÁ
    - vii. Rendelések felvétele és kezelése – BÁ
    - viii. Számlák generálása rendeléshez – BÁ
    - ix. Nyitvatartási órák kezelése – OK
    - x. Figyelmeztetések kezelése – OK
  - c. Triggerek:
    - i. Törzsvásárlóvá válás, ha a megrendelő legalább 5 rendelést leadott (tehát a hatodik rendeléstől). – BÁ
    - ii. Törzsvásárlói státusz elveszítése, ha a felhasználó nem lép be 3 hónap (90 nap!) leforgása alatt – BÁ
    - iii. Jelzés, ha egy kívánságlistás könyv akciós lesz - OK
    - iv. Értesítés küldése, ha egy számla több mint 2 napja nem lett kifizetve (felhasználói bejelentkezéshez kötött). – BÁ
    - v. Számla generálásnál, ha a vásárló törzsvásárló, 10%-al olcsóbb lesz a rendelés értéke. – BÁ
    - vi. Értesítés küldése, ha egy számla fizetése sikeres. - BÁ
  - d. Tárolt eljárások/függvények:
    - i. Egy könyv kategóriába sorolása készlet alapján (készleten, kevés van < 5, elfogyott) – BÁ
    - ii. Számla hovatartozásának az ellenőrzése. Input: számla id, felhasználó id, output: a felhasználó rendeléséhez tartozik-e a számla. – BÁ

- iii. Áruház besorolása nagysági kategóriákba, az összes készleten levő könyv darabszáma alapján. – BÁ
  - iv. Azok könyveknek a száma, melyből még egy példány sem került eladásra. – BÁ
  - v. Havi bevétel kiszámítása, input: melyik hónap (megfelelő alkalmazásoldali megvalósítással), output bevétel. - BÁ
5. Végegesítés  
Esetleges hibák javítása.

## Értékelési mód:

Csapat

## Feladat szöveges leírása

A könyvesboltlánc nagyléptékű bővülésével szükségessé vált egy webes felület kialakítása. Az a mi célunk, hogy ezt megtervezzük és megvalósítsuk.

A felmérések alapján nagy forgalmú lesz a weboldal, így az Oracle adatbázis-kezelő rendszer mellett döntöttünk.

Az oldalt bárki látogathatja majd, megtekintheti kínálatunkat, viszont a vásárlás csak egy gyors regisztráció után lehetséges. A visszatérő vásárlóinkat bőkezűen jutalmazzuk kedvezményekkel, amik az általános leárazások mellett érhetők el. Lehetőségük van a vásárlóknak könyveket kívánságlistára tenni, ami értesít, ha a választott könyv újra kapható lesz, vagy le lett árazva.

Célunk az is, hogy a vásárlóinkat minél több könyvvel ismertessük meg. A főoldalon minden megtekinthetők a legfrissebb kiadású könyvek, a könyvek oldalán pedig más vásárlókra alapozva ajánlunk egyéb könyveket. Készítünk heti és havi toplistát is a legkelendőbb könyvekből, illetve biztosítunk egy remek kereső felületet is.

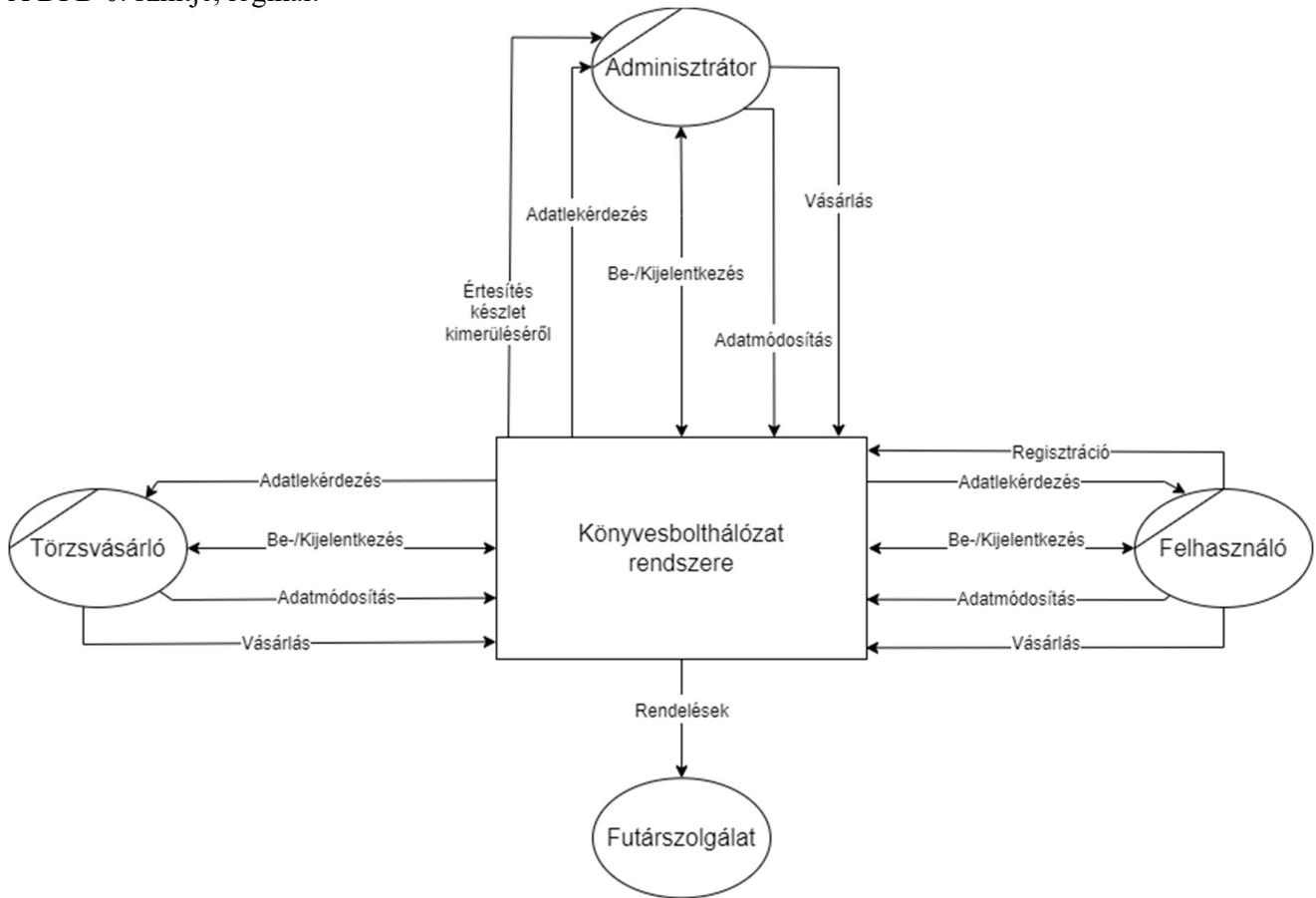
## Követelménykatalógus

- Könyvek felvétele és kezelése az adatbázisban (kiadás dátuma, kiadó, szerző, oldalszám, kötés, méret, ár)
- Műfajok és alműfajok kezelése és hozzáadása könyvekhez
- Műfajok mellé kigyűjteni, hogy mennyi könyv van az adott műfajban
- A legnépszerűbb könyvek műfajonként
- Keresés címre, szerzőre, műfajra, találatok számának megjelenítése
- Olcsó könyvek listázása
- Kedvezmények könyvekre
- Áruházak kezelése -> stock feltöltése
- Adminisztrátor, aki kezeli az adatokat (könyv, műfaj, áruházak, kedvezmények...)
- Mely könyvek kaphatók egy áruházban? Egy adott könyv mely áruházban kapható?
- Készlet nyilvántartása boltonként
- Figyelmeztetés készlet kimerüléséről
- Könyv vásárlása
- Csak regisztrált felhasználó tud vásárolni
- Kiszállítás vagy átvétel boltban opciók
- Számla készítése
- A legújabb könyvek megjelenítése
- Egy könyv adatlapjánál azon más könyveket is kilistázni, amelyeket megvettek azok a vásárlók, akik az aktuális könyvet megvették
- Heti/havi toplista a vásárolt könyvekről
- Törzsvásárlók nyilvántartása, törzsvásárlói kedvezmények
- Törzsvásárlóvá válás

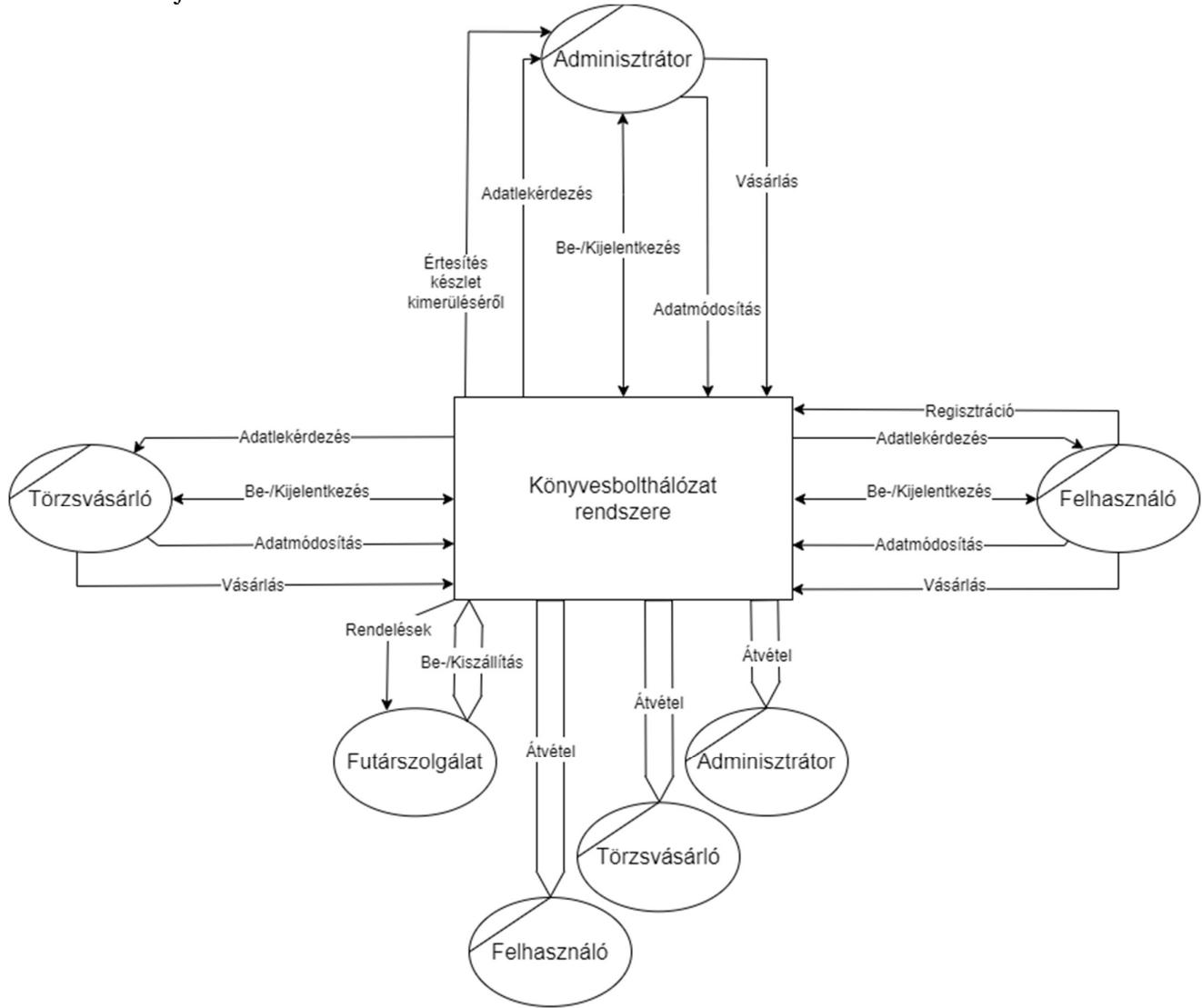
- Törzsvásárlói státusz elveszítése, ha a felhasználó nem lép be 3 hónap leforgása alatt
- Könyvek felvétele kívánságlistára
- Jelzés, ha egy kívánságlistás könyv akciós lesz
- Jelzés, ha egy kívánságlistás könyv újra kapható lesz

### **Adatfolyam diagram (DFD):**

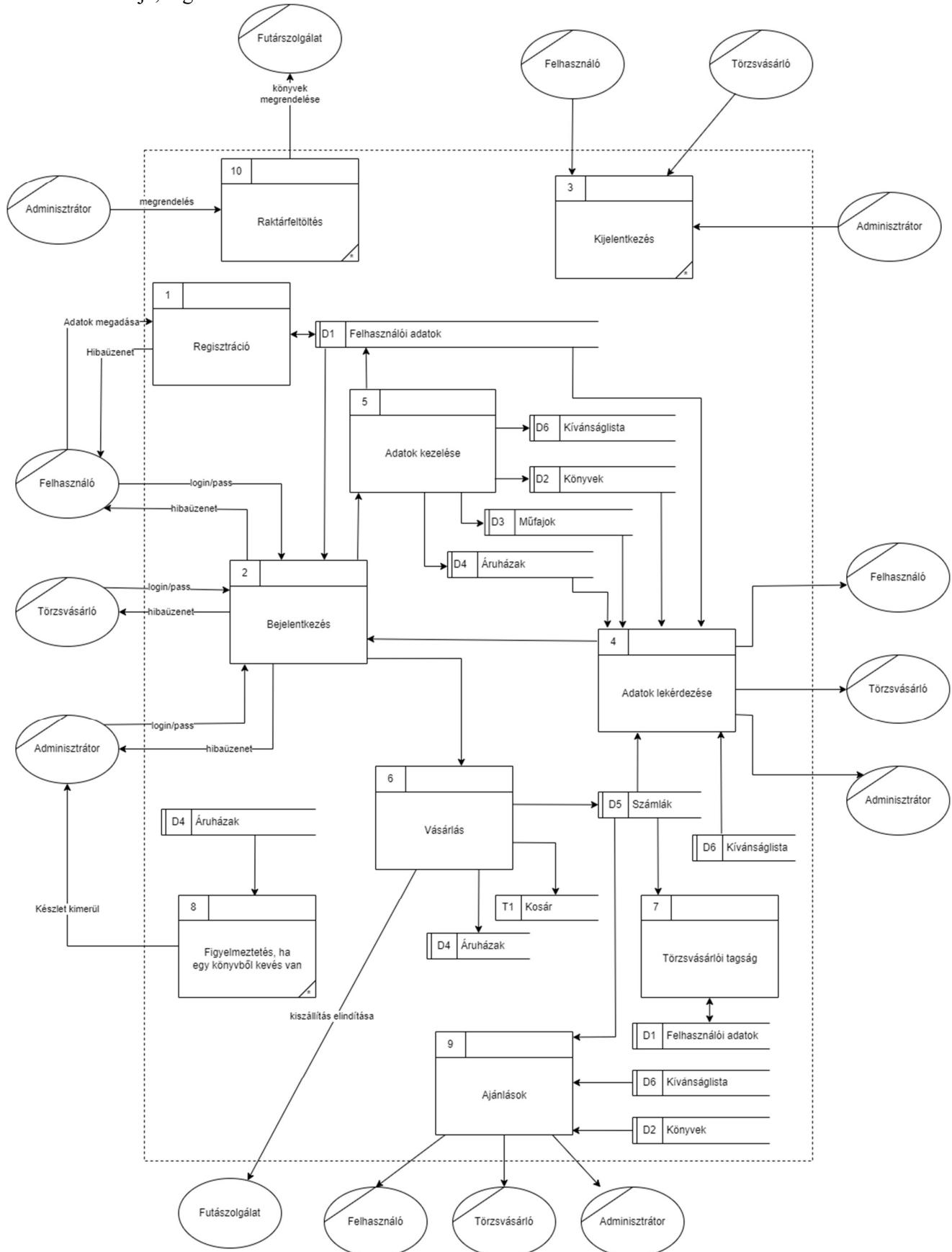
A DFD 0. szintje, logikai:



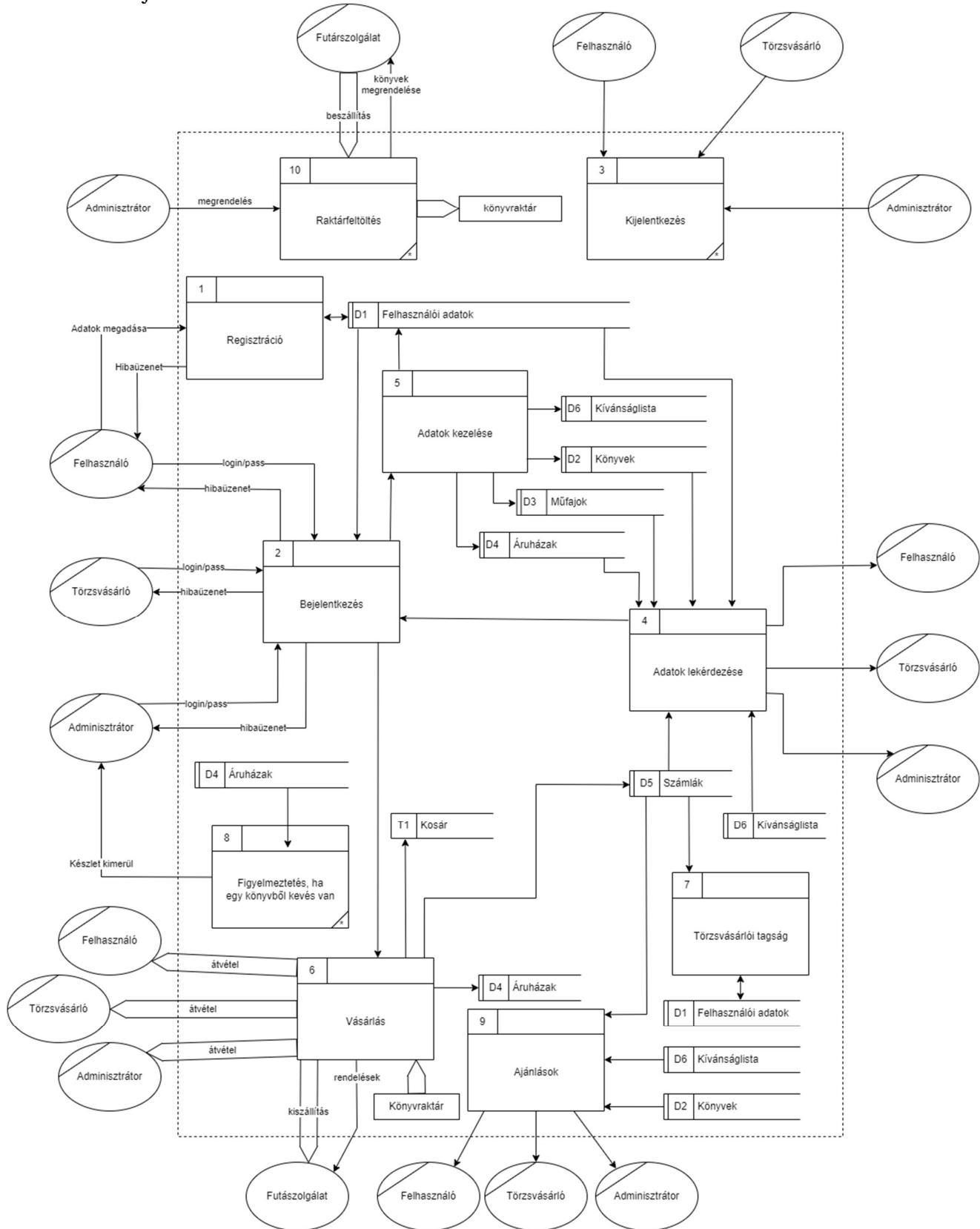
A DFD 0. szintje:



A DFD 1. szintje, logikai:

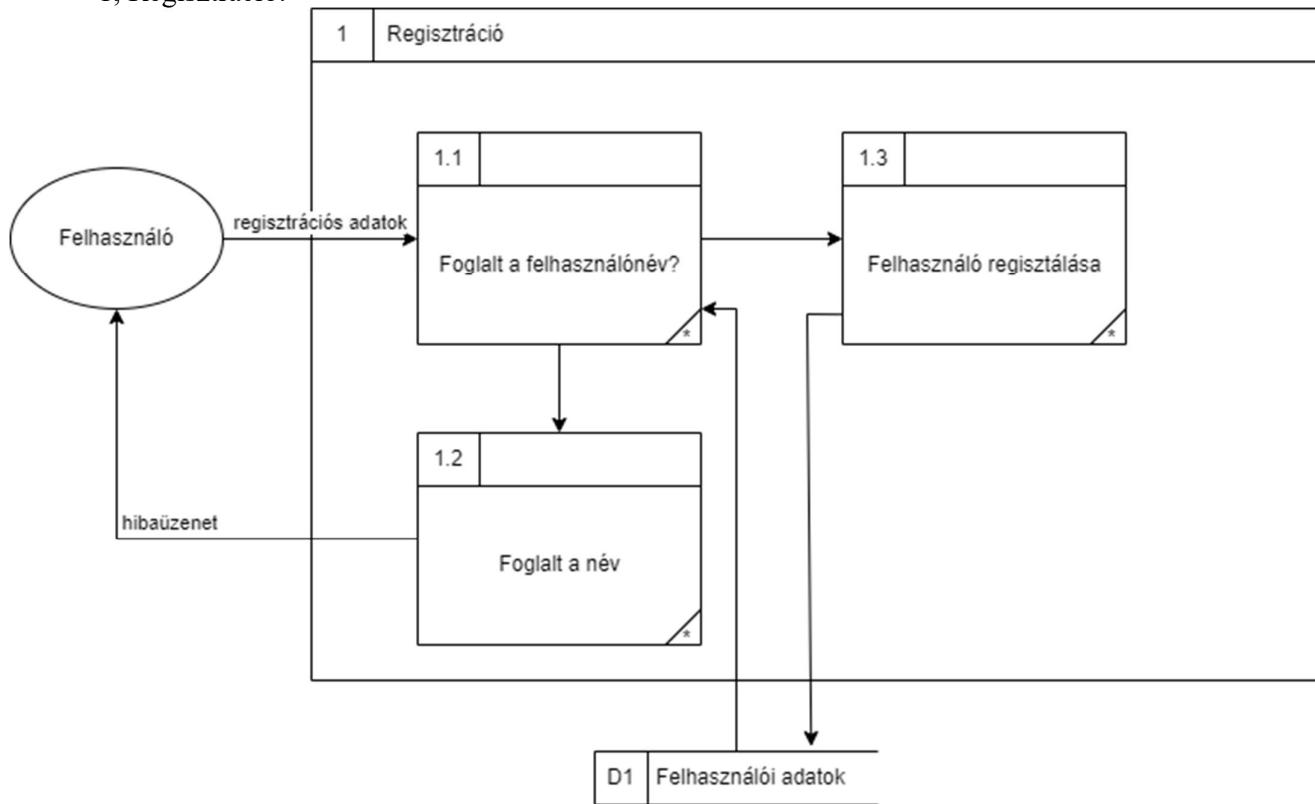


A DFD 1. szintje:

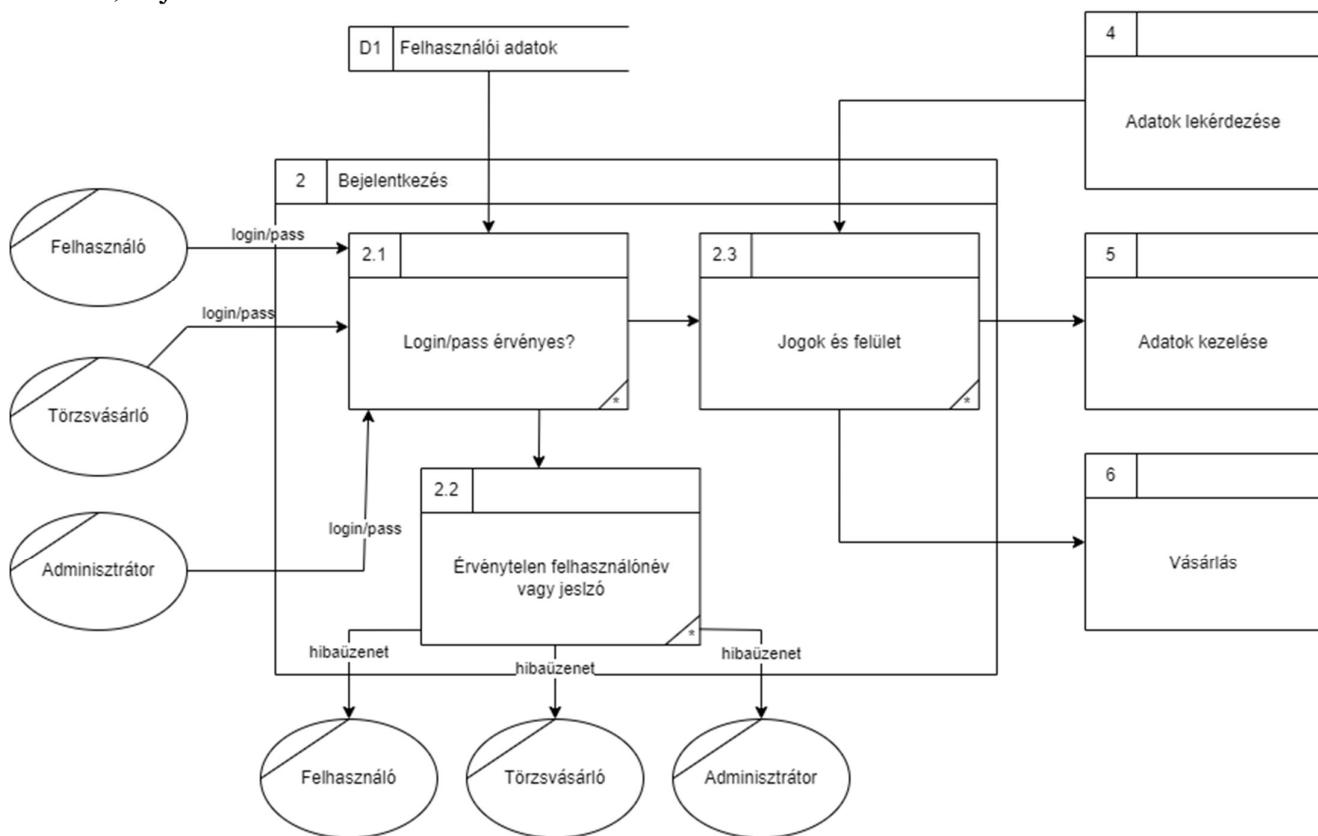


A DFD 2. szintje:

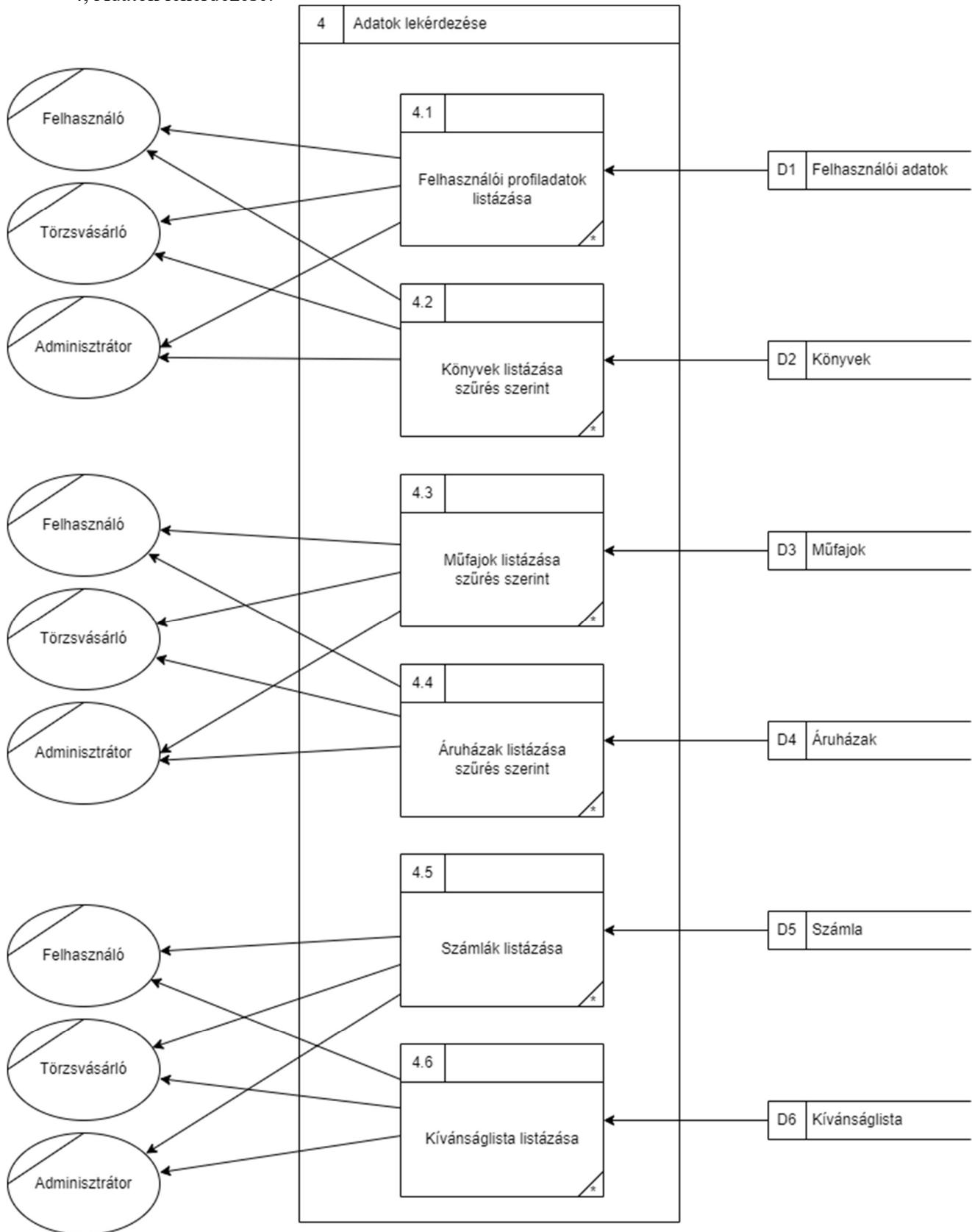
1, Regisztráció:



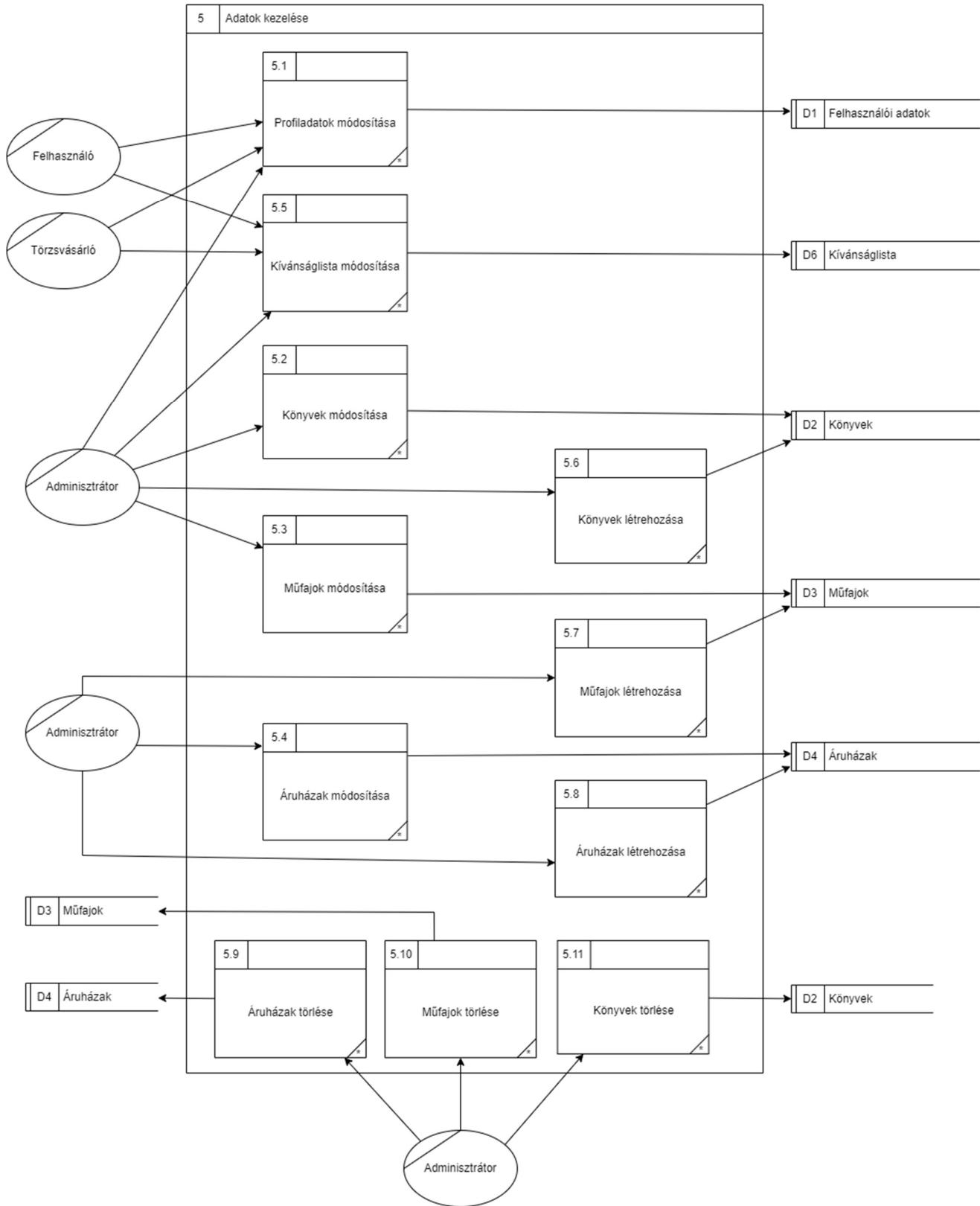
2, Bejelentkezés:



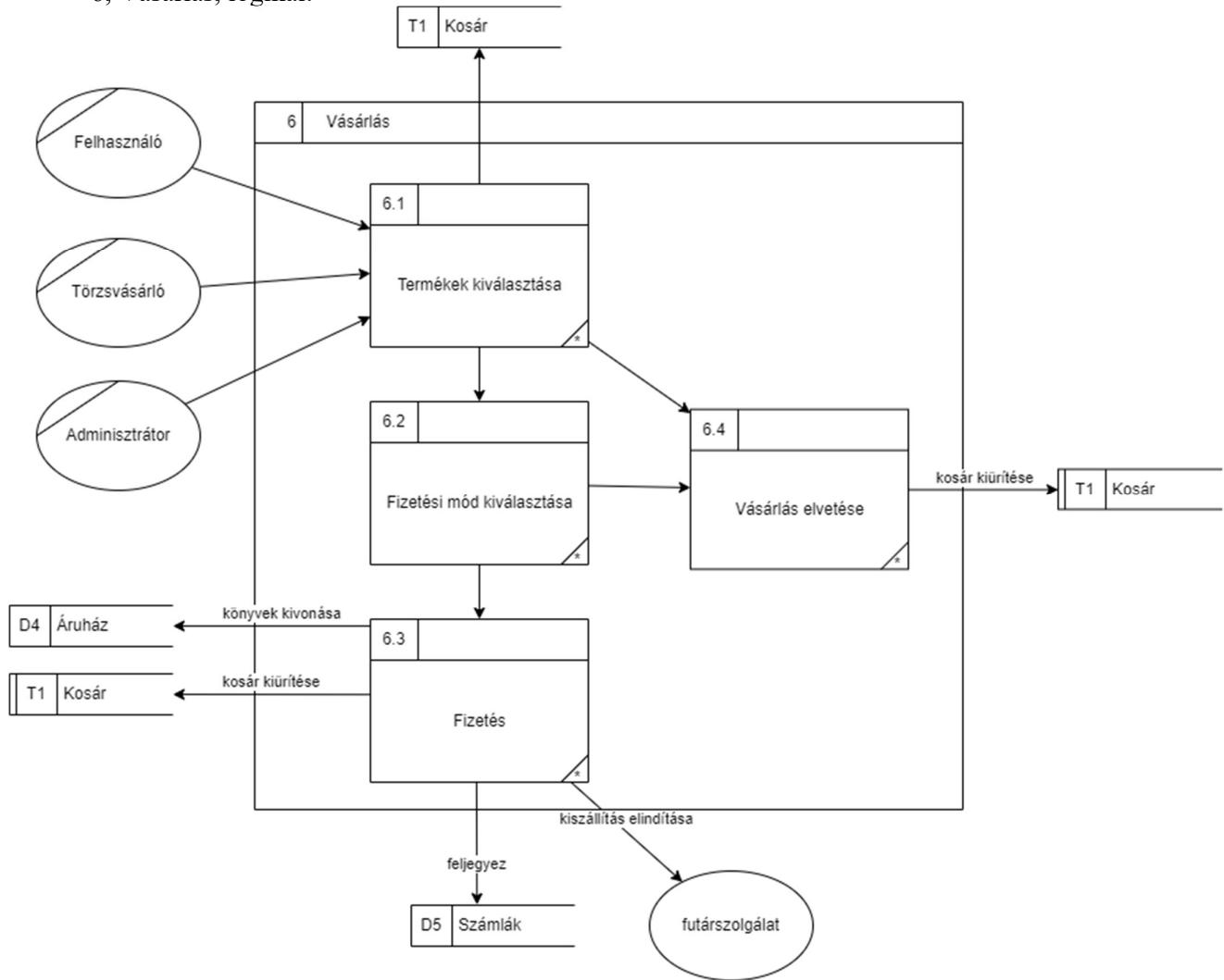
4, Adatok lekérdezése:



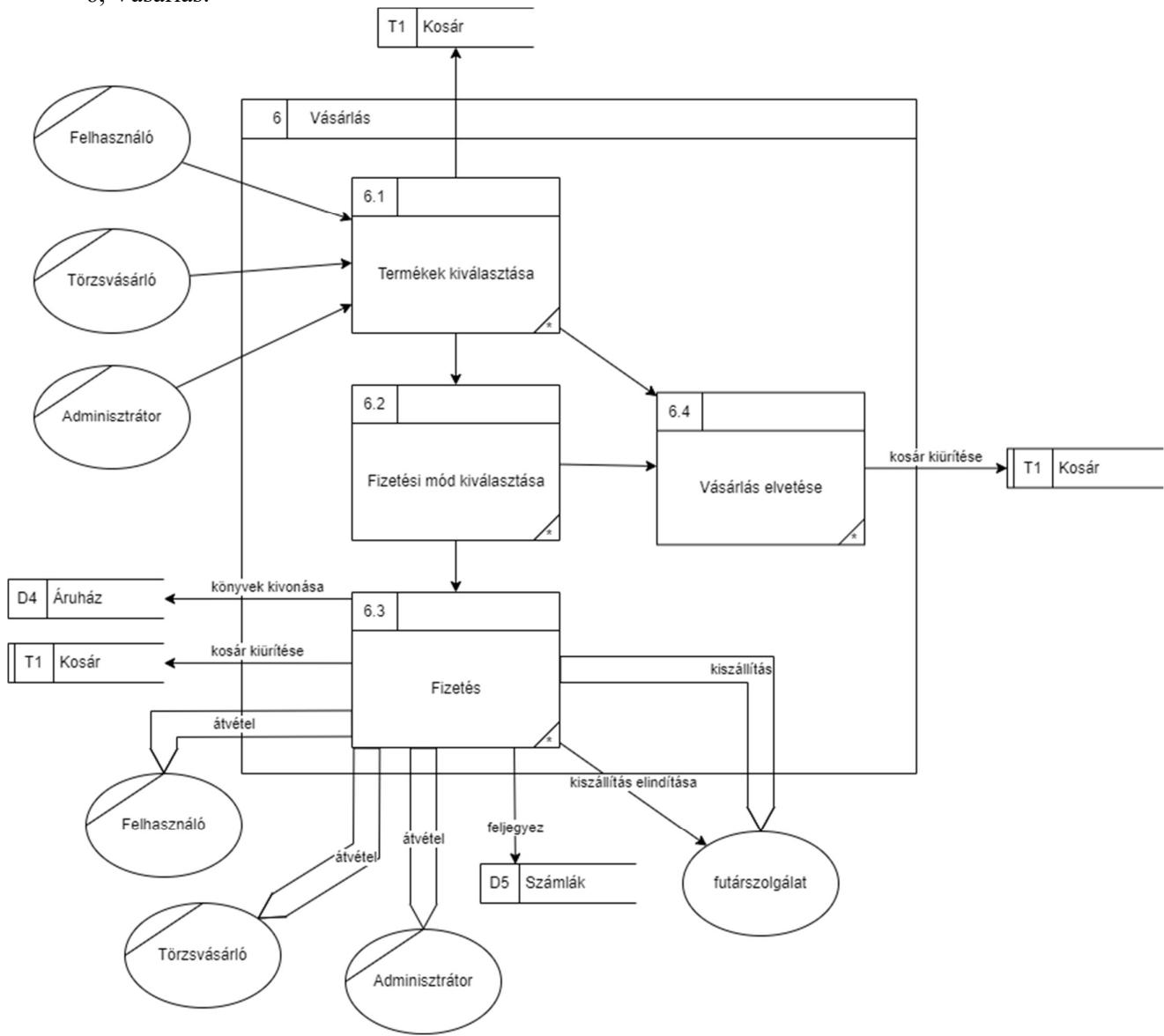
## 5. Adatok kezelése:



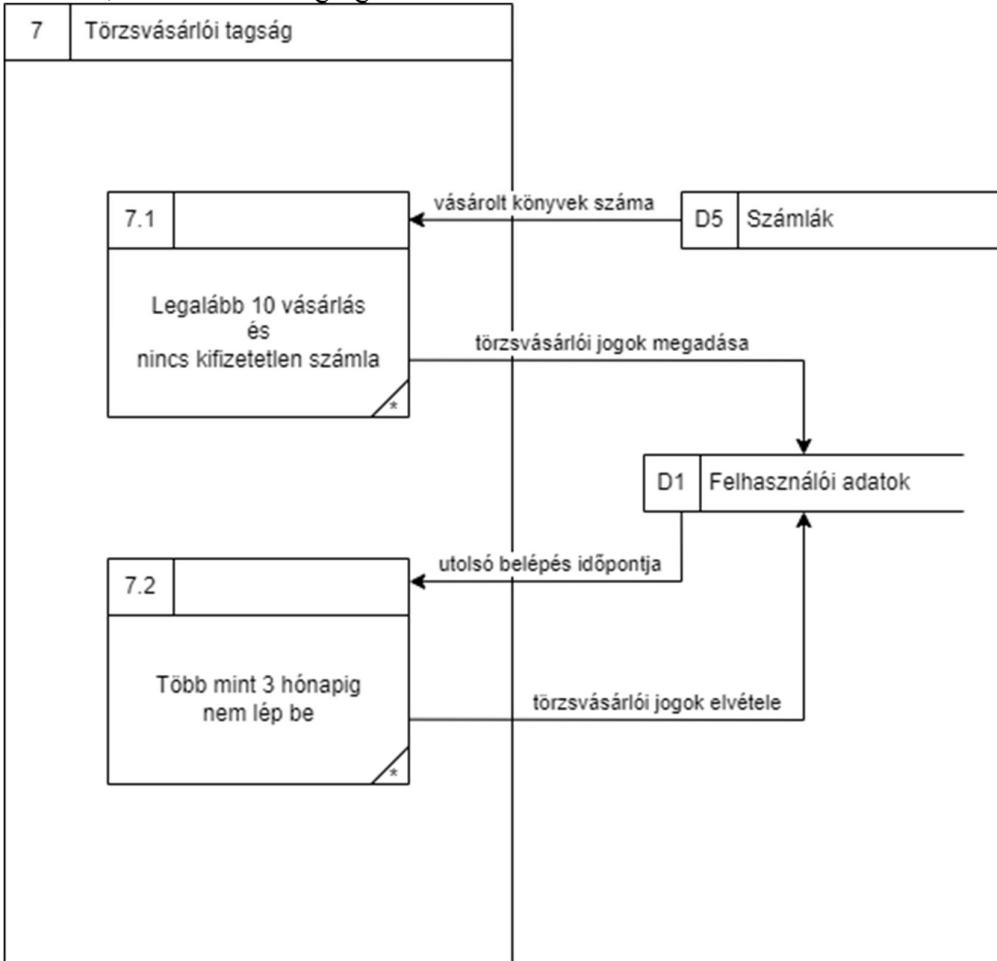
6, Vásárlás, logikai:



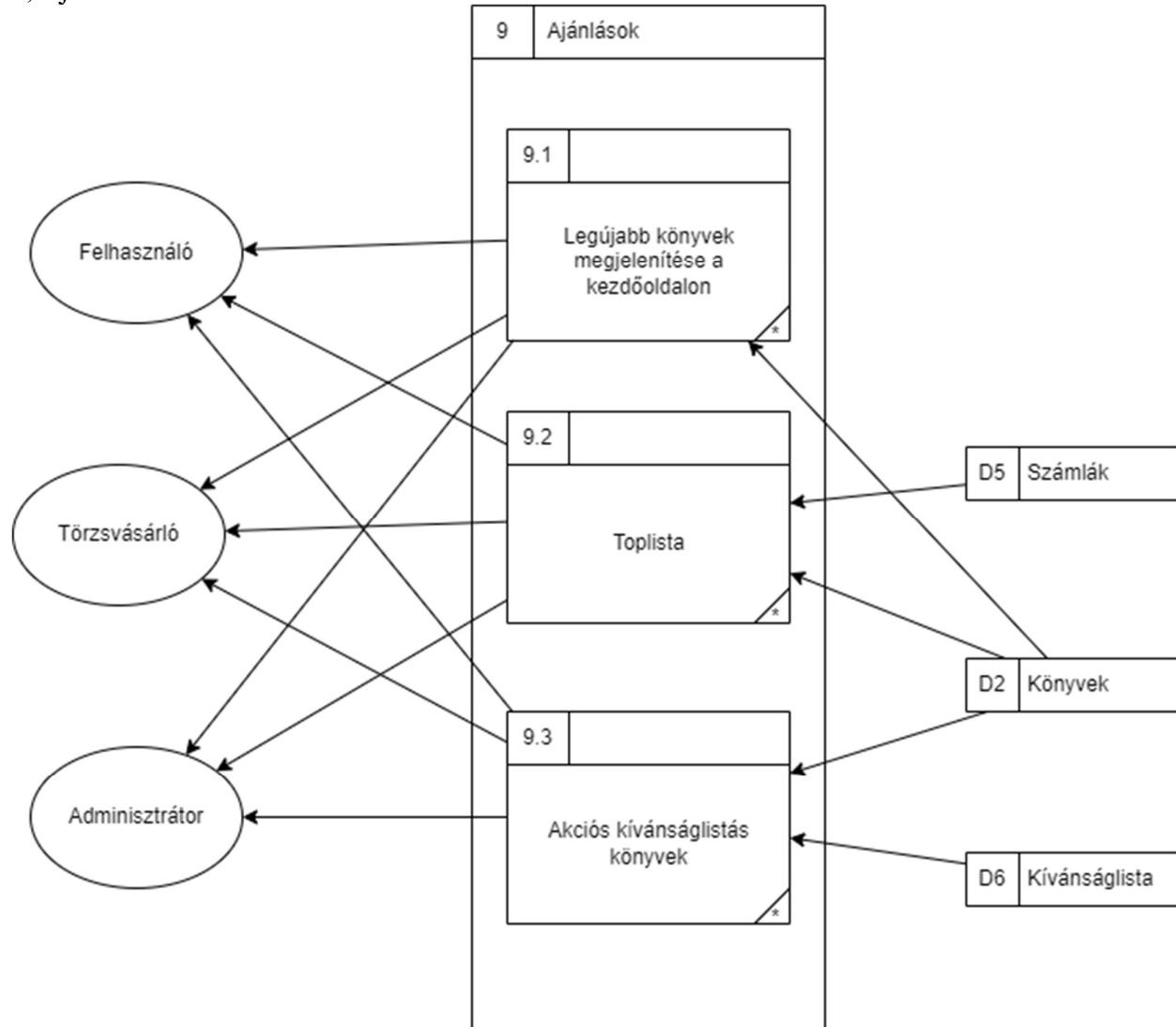
## 6, Vásárlás:



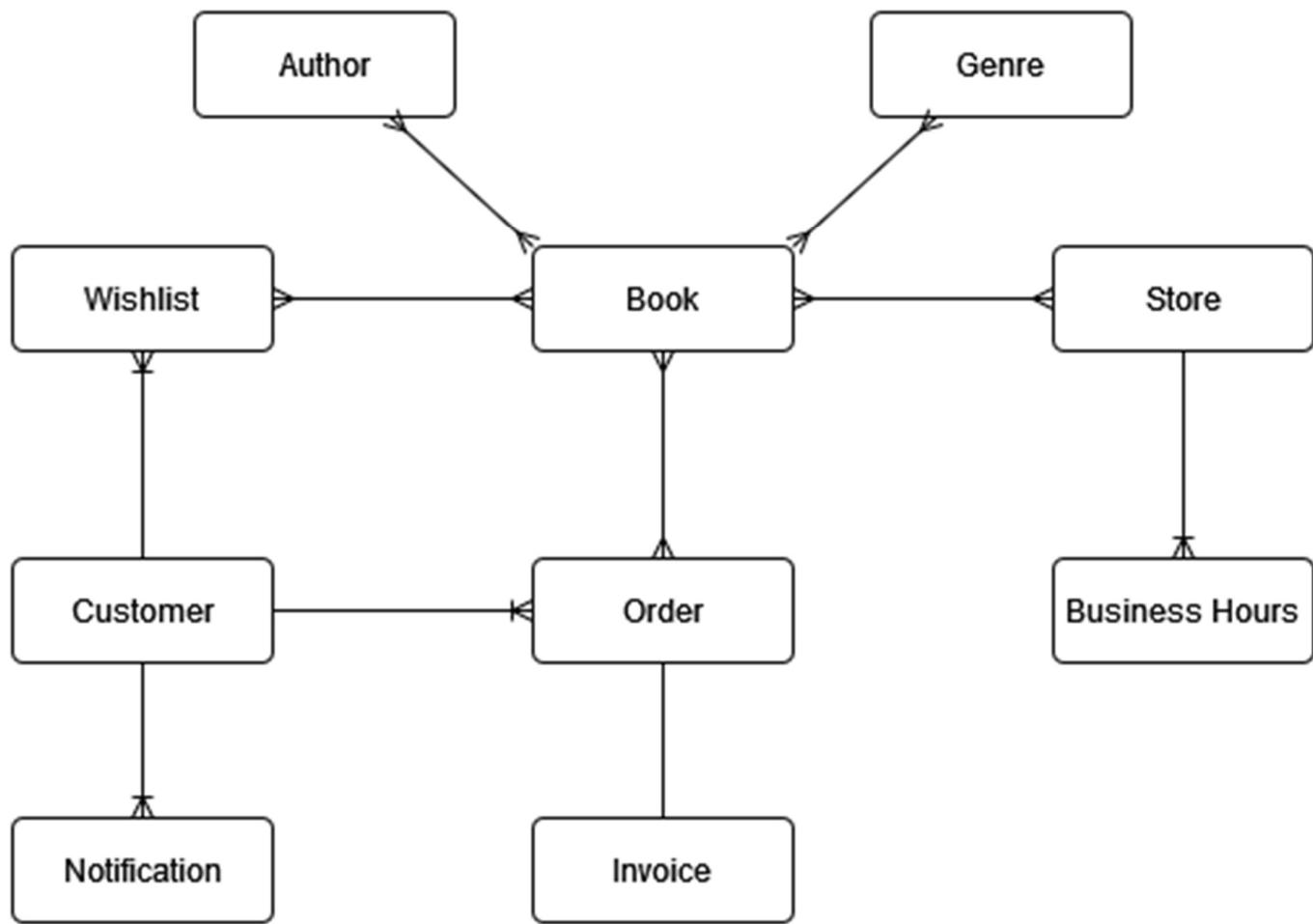
7. Törzsvásárlói tagság:



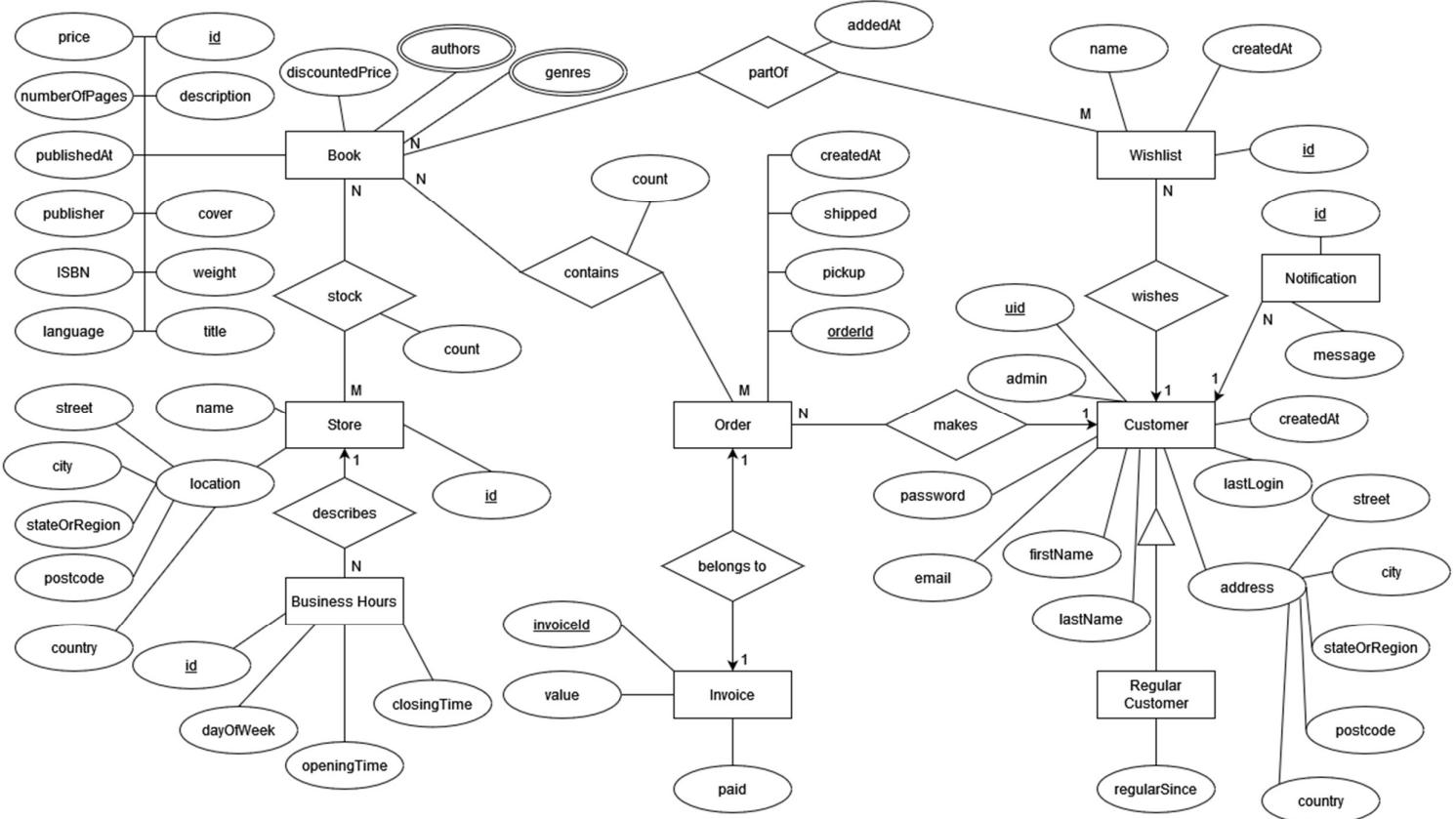
9, Ajánlások:



## Egyedmodell:



## EK-diagram:



( raw: <https://raw.githubusercontent.com/a-bakk/bookstore-oracle/master/docs/entity-relationship/er.drawio.png> )

### Az EK-diagram leképezése relációsémákra:

CUSTOMER (customerId, email, password, firstName, lastName, createdAt, lastLogin, admin, street, city, stateOrRegion, postcode, country, regularSince)

WISHLIST (wishlistId, name, createdAt, *customerId*)

ORDER (orderId, createdAt, shipped, pickup, *customerId*)

INVOICE (invoiceId, value, paid, *orderId*)

BOOK (bookId, title, description, cover, weight, price, numberOfPages, publishedAt, publisher, ISBN, language, discountedPrice)

AUTHOR (bookId, firstName, lastName)

GENRE (bookId, genreName)

STORE (storeId, name, street, city, stateOrRegion, postcode, country)

STOCK (bookId, storeId, count)

PARTOF (bookId, wishlistId, addedAt)

CONTAINS (*orderId*, *bookId*, count)

BUSINESS\_HOURS (*hoursId*, dayOfWeek, openingTime, closingTime, *storeId*)

NOTIFICATION (*notificationId*, message, *customerId*)

### A relációsémák normalizálása:

#### 1NF

Az összes séma megfelel az első normálforma előírásainak, hiszen leképezés után minden attribútum atomi.

#### 2NF

A következő sémkban minden kulcs egy attribútumból áll, ezért lesznek 2NF-ben: Customer, Wishlist, Order, Invoice, Book, Store, Business\_Hours, Notification.

A következő sémkban nincs másodlagos attribútum, ezért lesznek 2NF-ben: Author, Genre.

A maradék két sémában (Stock, part of és Contains) a kulcson kívüli egyetlen attribútum (count és addedAt) teljesen függ a kulcstól. (külön a bookId-tól nem függ a count, a storeId-tól szintén nem, és hasonlóan, a bookId-tól nem függ az addedAt és a wishlistId-tól sem külön)

#### 3NF

Az Author, Genre és Contains sémkban nincs másodlagos attribútum, ezért 3NF-ben vannak.

A Customer, Wishlist, Order, Invoice, Book, Store, Stock, PartOf, Business\_Hours és Notification sémák esetén kijelenthető, hogy nincs bennük tranzitív függés, ezért lesznek 3NF-ben.

Megjegyzés: Változtatna a helyzeten, ha az országok egyértelműen meghatároznák az irányítószámokat, [viszont ez nem teljesül](#). Ugyanilyen alapon jelenthető ki ez a városokról is.

## Táblák leírása:

**CUSTOMER:** Az ügyfelek tábla tárolja az ügyfelek adatait. Ugyanitt jelezhető, ha adott ügyfél adminisztrátor vagy törzsvásárló. A törzsvásárlói tagság elvesztéséhez az utolsó bejelentkezés időpontja is tárolásra kerül.

Név	Típus	Leírás
customer_id	NUMBER(19), PK, customer_seq	Az ügyfél egyedi azonosítója.
email	VARCHAR2(255), UNIQUE	Az ügyfél e-mail címe, szintén egyedi.
password	VARCHAR2(255)	Az ügyfél jelszava.
first_name	VARCHAR2(30)	Az ügyfél keresztnéve.
last_name	VARCHAR2(30)	Az ügyfél családneve.
created_at	DATE	Az ügyféli fiók létrehozásának dátuma.
last_login	DATE	Az ügyfél utolsó belépésének időpontja.
admin	NUMBER(1)	Tárolja, hogy adminisztrátor-e a felhasználó. 1, ha igen, 0, ha nem.
street	VARCHAR2(50)	Az ügyfél lakóhelyének/számlázási helyének az utcaneve és házszáma.
city	VARCHAR2(50)	Az ügyfél címének városa.
state_or_region	VARCHAR2(50)	A megye/régió megnevezése.
postcode	VARCHAR2(10)	A cím irányítószámának megnevezése. (*) irányítószámnál a leghosszabb postcode 10 jegy hosszú (Iráni Iszlám Köztársaság), míg az amerikai ZIP+4 rendszerben 9 jegy és egy kötőjel
country	VARCHAR2(56)	Az ország megnevezése. (*) a leghosszabb ország hivatalos neve The United Kingdom of Great Britain and Northern Ireland (56 hosszú)
regular_since	DATE	Az ügyfél törzsvásárlói tagságának kezdete.

**WISHLIST:** A kívánságlisták tárolására létrehozott tábla. Több könyvet is hozzá lehet adni egy nevesített listához, a hozzáadás időpontja szintén rögzítve van.

Név	Típus	Leírás
wishlist_id	NUMBER(19), PK, wishlist_seq	A kívánságlista egyedi azonosítója.
name	VARCHAR2(256)	A kívánságlistának adott név.

created_at	DATE	A kívánságlista létrehozatalának az időpontja.
customer_id	NUMBER(19)	A létrehozó ügyfél egyedi azonosítója.

ORDER: A rendelések tárolására alkalmas tábla. Alapvető információkat tárol a rendelésről, például, hogy mikor lett leadva és ki lett-e már szállítva. Ezek mellett az is megjelölhető, hogy áruházból átvételre szánt vagy szállítandó rendelés.

Név	Típus	Leírás
order_id	NUMBER(19), PK, order_seq	A rendelés egyedi azonosítója.
created_at	DATE	A rendelés leadásának dátuma.
shipped	NUMBER(1)	A rendelés át lett-e adva a futárcégnek, ha szállítást igényel.
pickup	NUMBER(1)	A rendelést a boltbol veszik át vagy szállítást igényel.
customer_id	NUMBER(19)	A rendelést leadó ügyfél azonosítója.

INVOICE: A számlák tárolására kialakított tábla. Itt már megjelenik a fizetendő összeg is, az, hogy készpénzzel vagy kártyával kerül kifizetésre, illetve hogy ki lett-e már fizetve.

Név	Típus	Leírás
invoice_id	NUMBER(19), PK, invoice_seq	A számla egyedi azonosítója.
value	NUMBER(12)	A számla értéke (HUF).
paid	NUMBER(1)	Ki van-e már fizetve a számla.
order_id	NUMBER(19)	Melyik rendeléshez tartozik a számla.

BOOK: A könyveket tároló tábla. Számos részlet van a könyvekről modellezve, érdemes talán megemlíteni, hogy az ISBN unique. Az áron kívül létezik egy leszállított, akciós ár is, hogy a kettő közötti különbséget is ki lehessen a későbbiekben számolni.

Név	Típus	Leírás
book_id	NUMBER(19), PK, book_seq	A könyv egyedi azonosítója.
title	VARCHAR2(128)	A könyv címe.
description	VARCHAR2(2048)	A könyv leírása.
cover	VARCHAR2(50)	A könyv borítójának típusa.

weight	FLOAT	A könyv súlya grammiban kifejezve.
price	NUMBER(10)	A könyv ára (HUF).
number_of_pages	NUMBER(5)	A könyv oldalainak száma.
published_at	DATE	A könyv kiadásának dátuma.
publisher	VARCHAR2(50)	A könyv kiadója.
ISBN	VARCHAR2(13)	A könyv ISBN száma, egyedi.
language	VARCHAR2(16)	A könyv nyelve.
discounted_price	NUMBER(10)	A könyv leárazott ára, ha létezik (HUF).

AUTHOR: A könyvek szerzőjét hivatott tárolni, tekintettel arra, hogy egy könyvhöz több szerző is tartozhat, de egy szerző több könyvet is írhat. A kereszt- és családnév szolgáltatja az értékes adatot.

Név	Típus	Leírás
book_id	NUMBER(19), PK	A könyv egyedi azonosítója, mely a szerzőhöz tartozik.
first_name	VARCHAR2(30), PK	A szerző keresztneve.
last_name	VARCHAR2(30), PK	A szerző családneve.

GENRE: A műfajokat rögzítő tábla. A külső kulcson kívül kizárolag a műfaj megnevezését tárolja.

Név	Típus	Leírás
book_id	NUMBER(19), PK	A könyv, melynek a műfaját tároljuk.
genre_name	VARCHAR2(30), PK	A műfaj megnevezése.

STORE: A könyváruházak adatait tároló tábla. A megnevezést és a címet tárolja.

Név	Típus	Leírás
store_id	NUMBER(19), PK, store_seq	Az áruház azonosítója.
name	VARCHAR2(50)	Az áruház megnevezése.
street	VARCHAR2(50)	Az áruház utcájának és házszámának megnevezése.
city	VARCHAR2(50)	Az áruház városa.
state_or_region	VARCHAR2(50)	A megye/régió megnevezése.

postcode	VARCHAR2(10)	Az áruház irányítószáma.
country	VARCHAR2(56)	Az áruház országa.

STOCK: Adott áruházból készleten levő könyveket tárolja, pontosabban a darabszámot.

Név	Típus	Leírás
book_id	NUMBER(19), PK	A könyv azonosítója, melynek áruházi készletét jelöljük.
store_id	NUMBER(19), PK	Az áruház azonosítója, melyben a könyv adott számban megtalálható.
count	NUMBER(6)	Adott könyvből hány darab van az áruházban.

PARTOF: Tárolja, hogy mely könyvek vannak adott kívánságlistán. Ezen kívül az is megjelenik, hogy mikor lett hozzáadva.

Név	Típus	Leírás
book_id	NUMBER(19), PK	A könyv azonosítója, melyet kívánságlistára helyezünk.
wishlist_id	NUMBER(19), PK	A kívánságlista azonosítója, melyre a könyvet helyezzük.
addedAt	DATE	A hozzáadás időpontja.

CONTAINS: Adott rendeléshez mely könyvek tartoznak, a könyvek és rendelések kapcsolatát leíró tábla.

Név	Típus	Leírás
order_id	NUMBER(19), PK	A rendelés azonosítója, melyhez több könyv tartozik.
book_id	NUMBER(19), PK	A könyv azonosítója, mely a rendeléshez tartozik.
count	NUMBER(6)	Hány példányt tartalmaz a rendelés.

BUSINESS\_HOURS: Az áruházak nyitvatartási ideje napra és órára bontva.

Név	Típus	Leírás
hours_id	NUMBER(19), PK, business_hours_seq	A nyitvatartási órák egyedi azonosítója.
day_of_week	NUMBER(1, 0)	A hét napjának azonosítója, hétfő = 1, ...
opening_time	VARCHAR2(30)	A nyitás órája.
closing_time	VARCHAR2(30)	A zárás órája.
store_id	NUMBER(19)	Melyik áruházhhoz tartoznak az adatok.

NOTIFICATION: A felhasználóknak küldött üzenetek, melyek jelezhetik például könyv kimerülését a készletből.

Név	Típus	Leírás
notification_id	NUMBER(19), PK, notification_seq	Az üzenet azonosítója.
message	VARCHAR2(256)	Az üzenet tartalma.
customer_id	NUMBER(19)	Melyik felhasználóhoz tartozik a figyelmeztetés.

## Szerep-funkció mátrix és Egyed-esemény mátrix és Funkció megadása

## Szerep-funkció mátrix:

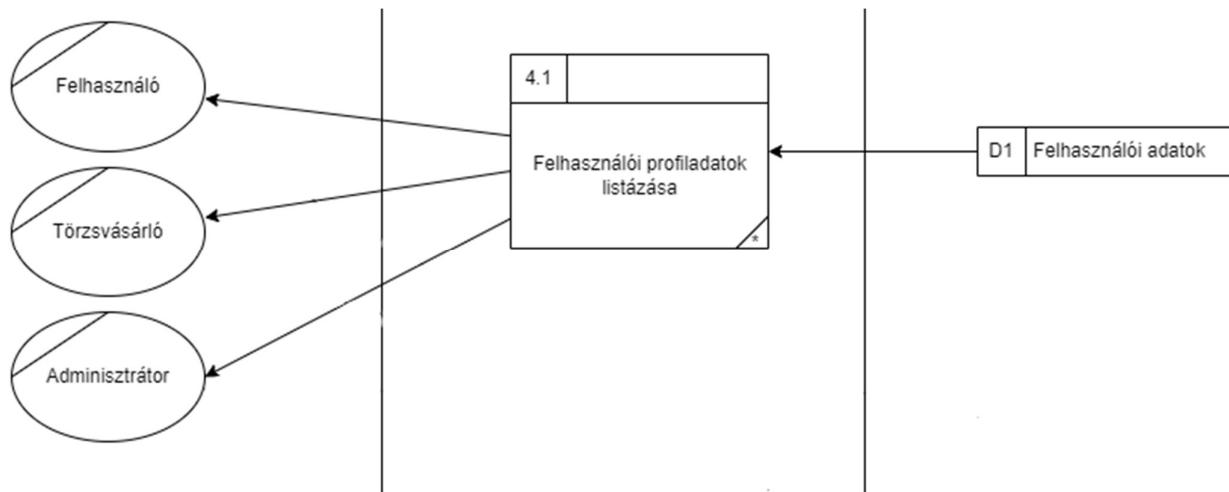
	Kijelentkezés	X	X	X
Vásárlás		X	X	X
Számlák lekérdezése/módosítása		X	X	X
Kívánságlista lekérdezése/módosítása		X	X	X
Áruházak lekérdezése	X	X	X	X
Műfajok lekérdezése	X	X	X	X
Könyvek lekérdezése	X	X	X	X
Áruházaik módosítása/törlése			X	X
Műfajok módosítása/törlése			X	X
Könyvek módosítása/törlése			X	X
Áruházaik létrehozása			X	X
Műfajok létrehozása			X	X
Könyvek létrehozása			X	X
Bejelentkezés				
Regisztráció	X	X		
Vendég				
Felhasználó				
Törzsvásárló				
Adminisztrátor				

### Egyed-esemény mátrix:

L: Létrehozás , M: Módosítás, O: Olvasás, T: Törlés

## Funkció megadása

Felhasználói profiladatok listázása



Funkciónév:

Típus:

Felhasználói szerepkörök:

Funkció leírás:

Hibakezelés:

AFD eljárások:

Köv. katalógusra hivatkozás:

Tömegszerűség:

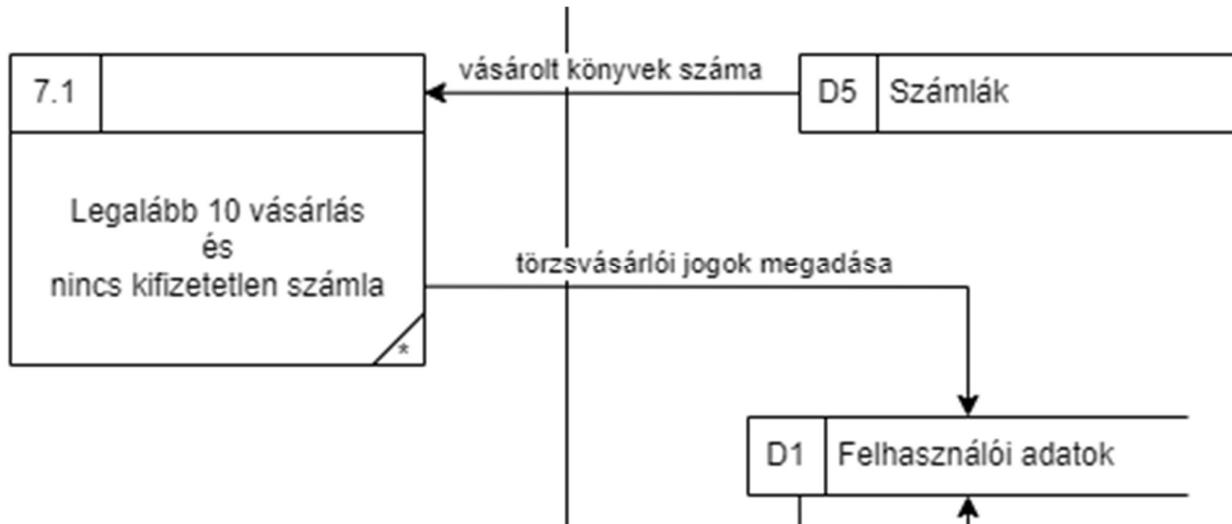
Lekérdezések:

Közös feldolgozás:

Válaszidő/tűrés:

Profiladatok listázása
lekérdezés, online, felhasználói
adminisztrátor, törzsvásárló, felhasználó
Adott felhasználó profiladatainak a lekérése.
Nincs adott azonosítójú felhasználó.
4.1
Regisztrált felhasználó adatainak módosítása, törlése.
kb. napi 150
adott azonosítójú felhasználó keresése, gyakorisága kb. napi 150
adott azonosítójú felhasználó keresése
0.5 sec / 5 sec

## Törzsvásárlóvá válás



Funkciónév:

Törzsvásárlóvá válás

Típus:

karbantartás, online, rendszerfunkció

Felhasználói szerepkörök:

felhasználó

Funkció leírás:

Felhasználó törzsvásárlóvá válása, ha teljesíti a feltételeket.

Hibakezelés:

Nincs adott azonosítójú felhasználó.

AFD eljárások:

7.1

Köv. katalógusra hivatkozás:

Olcsóbb vásárlási lehetőségek.

Tömegszerűség:

kb. napi 2

Lekérdezések:

adott azonosítójú felhasználó keresése, módosítása, gyakorisága kb. napi 2

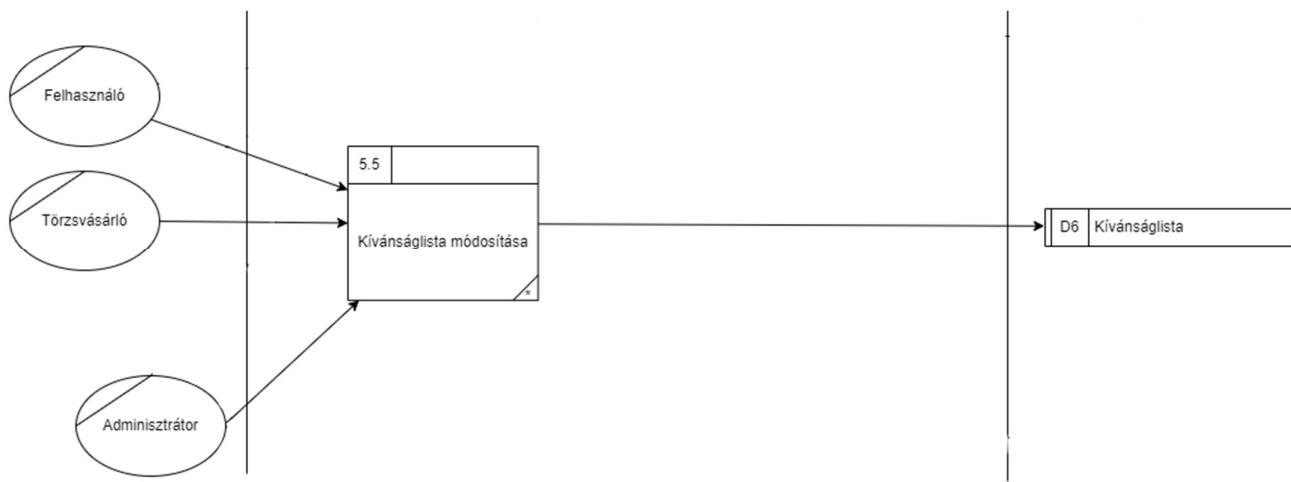
Közös feldolgozás:

adott azonosítójú felhasználó keresése, módosítása

Válaszidő/tűrés:

3 sec / 15 sec

## Kívánságlista módosítása



Funkciónév:

Kívánságlista módosítása

Típus:

karbantartás (leérdezést is tartalmaz), online, felhasználói

Felhasználói szerepkörök:

felhasználó, törzsvásárló

Funkció leírás:

A kívánságlista elemeinek törlése, új hozzaadása, esetleg ürítés.

Hibakezelés:

Nincs adott azonosítójú kívánságlista.

AFD eljárások:

5.5

Köv. katalógusra hivatkozás:

Értesítést kaphat a felhasználó, ha akciós.

Tömegszerűség:

kb. napi 30

Lekérdezések:

adott azonosítójú felhasználó keresése, kívánságlista keresése és módosítása, gyakorisága kb. napi 30

Közös feldolgozás:

adott azonosítójú felhasználó keresése, kívánságlista keresése és módosítása

Válaszidő/tűrés:

1.5 sec / 5 sec

## Összetett lekérdezések

Az alkalmazás forráskódjában az összetett lekérdezéseket megvalósító metódusok JavaDoc formájában meg lettek jelölve: [Összetett lekérdezés]. Ez alapján projektszintű kereséssel könnyen megtalálhatóak a megvalósítások (IntelliJ Idea-ban pl. CTRL+SHIFT+F).

1) A legnépszerűbb írók meghatározása rendelések alapján. AuthorDao.java::34

```
SELECT a.first_name, a.last_name
FROM contains c
JOIN author a ON c.book_id = a.book_id
GROUP BY a.first_name, a.last_name
ORDER BY SUM(c.count) DESC;
```

2) Az ügyfelek és a rendelései számának lekérdezése. CustomerDao.java::53

```
SELECT c.customer_id, c.first_name, c.last_name, COUNT(*) AS number_of_orders
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY number_of_orders DESC;
```

3) Meghatározza, hogy melyik ügyfél rendelt a legutóbb. CustomerDao.java::84

```
SELECT c.first_name, c.last_name
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
WHERE o.created_at = (SELECT MAX(newest.created_at) FROM orders newest)
ORDER BY c.customer_id DESC
FETCH FIRST ROW ONLY;
```

4) Azon 3 szerző lekérdezése, melyeknek a legdrágábbak a könyvei (az átlagár a legmagasabb). AuthorDao.java::63

```
SELECT a.first_name, a.last_name, AVG(b.price)
FROM author a
JOIN book b ON b.book_id = a.book_id
GROUP BY a.first_name, a.last_name
ORDER BY AVG(b.price) DESC
FETCH FIRST 3 ROWS ONLY;
```

5) Könyvajánlások lekérdezése: ha egy könyv szerepel egy rendelésben, lekérdezzük az ugyanazokban a rendelésekben szereplő más könyveket -> egy könyvhöz azokat ajánljuk, amelyet velük együtt rendeltek. Paraméterezett lekérdezés, inputja egy könyv/könyv id. BookDao.java::239

```
SELECT b.book_id, b.title
FROM contains c1
JOIN contains c2 ON c1.order_id = c2.order_id
JOIN book b ON b.book_id = c2.book_id
WHERE c1.book_id = 1 AND c2.book_id != 1
GROUP BY b.book_id, b.title;
```

6) minden üzlethez lekérdezzük a raktáron levő könyvek számát. StoreDao.java::86

```
SELECT s.store_id, s.name, SUM(st.count)
FROM store s
JOIN stock st ON st.store_id = s.store_id
GROUP BY s.store_id, s.name
ORDER BY SUM(st.count) DESC;
```

7) A három legnépszerűbb könyvet kérdezzük le, rendelések számának alapján. BookDao.java::257

```
SELECT b.book_id, b.title, COUNT(*) AS order_count
FROM book b
JOIN contains c ON b.book_id = c.book_id
GROUP BY b.book_id, b.title
ORDER BY order_count DESC
FETCH FIRST 3 ROWS ONLY;
```

8) A három legnépszerűbb könyvet kérdezzük le műfajonként. Paraméterezett lekérdezés, az input a műfaj neve. Ha 0 rendelés van adott kategóriában az összes könyvre, a rendelések számát figyelmen kívül hagyjuk. GenreDao.java::69

```
SELECT b.book_id, b.title, COUNT(*) AS order_count
FROM book b
JOIN genre g ON b.book_id = g.book_id
LEFT JOIN contains c ON b.book_id = c.book_id
WHERE g.genre_name = 'kaland'
GROUP BY b.book_id, b.title
ORDER BY order_count DESC
FETCH FIRST 3 ROWS ONLY;
```

9) A legnépszerűbb műfaj lekérdezése rendelések számának alapján. GenreDao.java::91

```
SELECT g.genre_name
FROM contains c
JOIN genre g ON c.book_id = g.book_id
GROUP BY g.genre_name
ORDER BY SUM(c.count) DESC
FETCH FIRST ROW ONLY;
```

10) A műfajonkénti átlagár kiszámítása. GenreDao.java::108

```
SELECT g.genre_name, AVG(b.price)
FROM book b
JOIN genre g ON g.book_id = b.book_id
GROUP BY g.genre_name
ORDER BY AVG(b.price) DESC;
```

11) Meghatározzuk mely három ügyfélnek van a legtöbb könyv a kívánságlistáján. WishlistDao.java::81

```
SELECT c.customer_id, c.first_name, c.last_name, COUNT(p.book_id) AS number_of_items
FROM customer c
JOIN wishlist w ON c.customer_id = w.customer_id
JOIN partof p ON w.wishlist_id = p.wishlist_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY number_of_items DESC
FETCH FIRST 3 ROWS ONLY;
```

## Tárolt eljárások/függvények

Hasonlóan a lekérdezésekhez, az eljárások hívásának helyét a programkódban [Tárolt eljárás] JavaDoc komment jelöli. Az alkalmazásból kizártlag eljárások vannak hívva, hiszen a JPA nem támogatja a függvényeket, bizonyos workarounds alkalmazásával (lemenni a JDBC szintjére) talán sikerült volna, viszont az eljárások OUT paraméterei szintén megfelelőnek bizonyultak visszatérési értéknek.

1) Számla hovatartozásának az ellenőrzése. Input: számla id, felhasználó id, output: a felhasználó rendeléséhez tartozik-e a számla.

```

CREATE OR REPLACE PROCEDURE invoice_belongs_to_customer
(in_invoice_id IN NUMBER, in_customer_id IN NUMBER, out_result OUT NUMBER)
IS
    curr_order orders%ROWTYPE;
BEGIN

    SELECT o.order_id, o.created_at, o.shipped, o.pickup, o.customer_id
    INTO curr_order
    FROM orders o
    JOIN invoice i ON o.order_id = i.order_id
    WHERE i.invoice_id = in_invoice_id;

    IF curr_order.customer_id = in_customer_id THEN
        out_result := 1;
    ELSE
        out_result := 0;
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        out_result := 0;
END invoice_belongs_to_customer;
/

```

2) Egy könyv kategóriába sorolása készlet alapján (készleten, kevés van < 5, elfogyott).

```

CREATE OR REPLACE PROCEDURE stock_status_per_book
(in_book_id IN NUMBER, out_status OUT VARCHAR2)
IS
    number_of_books NUMBER;
BEGIN

    SELECT SUM(COALESCE(st.count, 0))
    INTO number_of_books
    FROM stock st
    WHERE st.book_id = in_book_id
    GROUP BY st.book_id;

    IF number_of_books > 5 THEN
        out_status := 'ON_STOCK';
    ELSIF number_of_books > 0 THEN
        out_status := 'FEW_REMAINING';
    ELSE
        out_status := 'NONE';
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        out_status := 'NONE';
END stock_status_per_book;
/

```

3) Áruház besorolása nagysági kategóriákba, az összes készleten levő könyv darabszáma alapján.

```

CREATE OR REPLACE PROCEDURE store_size
(in_store_id IN NUMBER, out_status OUT VARCHAR2)
IS
    number_of_books NUMBER;
BEGIN

    SELECT SUM(COALESCE(st.count, 0))
    INTO number_of_books
    FROM stock st
    WHERE st.store_id = in_store_id
    GROUP BY st.store_id;

```

```

    IF number_of_books > 100 THEN
        out_status := 'LARGE_STORE';
    ELSIF number_of_books > 30 THEN
        out_status := 'MEDIUM_STORE';
    ELSE
        out_status := 'SMALL_STORE';
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        out_status := 'SMALL_STORE';
END store_size;
/

```

4) Azok könyveknek a száma, melyből még egy példány sem került eladásra.

```

CREATE OR REPLACE PROCEDURE unsold_books
(number_of_books OUT NUMBER)
IS
BEGIN

    SELECT COUNT(b.book_id)
    INTO number_of_books
    FROM book b
    WHERE b.book_id NOT IN (
        SELECT c.book_id
        FROM contains c
    );
END unsold_books;
/

```

5) Havi bevétel kiszámítása, input: melyik hónap (megfelelő alkalmazásoldali megvalósítással), output bevétel.

```

CREATE OR REPLACE PROCEDURE revenue_per_month
(in_start_date IN DATE, in_end_date IN DATE, out_result OUT NUMBER)
IS
BEGIN

    SELECT SUM(i.value)
    INTO out_result
    FROM invoice i
        JOIN orders o ON i.order_id = o.order_id
    WHERE o.created_at >= in_start_date
        AND o.created_at < in_end_date;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        out_result := 0;
END revenue_per_month;
/

```

6) Egy ügyfélhez tartozó rendelések számának meghatározása (trigger használja).

```

CREATE OR REPLACE FUNCTION count_orders_for_customer
(in_customer_id IN NUMBER) RETURN NUMBER
IS
    order_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO order_count
    FROM orders o
    WHERE o.customer_id = in_customer_id;
    RETURN order_count;

```

```

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN 0;
END count_orders_for_customer;
/

```

## Triggerek

1) Törzsvásárlóvá válás.

```

CREATE OR REPLACE TRIGGER become_regular_customer
AFTER INSERT ON contains
FOR EACH ROW
DECLARE
  curr_customer_id NUMBER;
BEGIN

  SELECT o.customer_id
  INTO curr_customer_id
  FROM orders o
  WHERE o.order_id = :NEW.order_id;

  UPDATE customer c
  SET c.regular_since = SYSDATE
  WHERE c.customer_id = curr_customer_id
    AND count_orders_for_customer(curr_customer_id) > 5
    AND c.regular_since IS NULL;
END;
/

```

2) Törzsvásárlói státusz elvesztése.

```

CREATE OR REPLACE TRIGGER lose_regular_status
BEFORE UPDATE OF last_login ON customer
FOR EACH ROW
BEGIN
  IF :OLD.last_login < (SYSDATE - 90) THEN
    :new.regular_since := NULL;
  END IF;
END;
/

```

3) A számla értékének újraszámítása, ha az ügyfél törzsvásárló.

```

CREATE OR REPLACE TRIGGER recalculate_value_if_regular
BEFORE INSERT ON invoice
FOR EACH ROW
DECLARE
  curr_customer customer%ROWTYPE;
  curr_customer_id NUMBER;
BEGIN
  SELECT o.customer_id
  INTO curr_customer_id
  FROM orders o
  WHERE o.order_id = :NEW.order_id;

  SELECT *
  INTO curr_customer
  FROM customer c
  WHERE c.customer_id = curr_customer_id;

  IF curr_customer.regular_since IS NOT NULL THEN
    :NEW.value := 0.9 * :NEW.value;
  END IF;
END;

```

/

4) Értesítés küldése, ha egy számla több mint 2 napon át nem volt kifizetve.

```
CREATE OR REPLACE TRIGGER notif_when_invoice_is_unpaid
AFTER UPDATE OF last_login ON customer
FOR EACH ROW
DECLARE
    CURSOR orders_for_curr_customer IS
        SELECT order_id, customer_id, created_at FROM orders WHERE orders.customer_id =
:NEW.customer_id;
        is_paid NUMBER(1);
BEGIN
    FOR ord IN orders_for_curr_customer
    LOOP
        SELECT i.paid
        INTO is_paid
        FROM invoice i
        WHERE i.order_id = ord.order_id;
        IF is_paid = 0 AND ord.created_at < SYSDATE - 2 THEN
            INSERT INTO notification(message, customer_id) VALUES
            ('2 napnál régebbi kifizetetlen számlája van!', ord.customer_id);
        END IF;
    END LOOP;
END;
/
```

5) Értesítés küldése a fizetés sikerességéről.

```
CREATE OR REPLACE TRIGGER notif_when_payment_successful
AFTER UPDATE OF paid ON invoice
FOR EACH ROW
DECLARE
    curr_customer_id NUMBER;
BEGIN
    SELECT o.customer_id
    INTO curr_customer_id
    FROM orders o
    WHERE o.order_id = :NEW.order_id;

    IF :OLD.paid = 0 AND :NEW.paid = 1 THEN
        INSERT INTO notification(message, customer_id) VALUES
        ('Sikeresen kifizettet egy számlát!', curr_customer_id);
    END IF;
END;
/
```

6) Értesítés küldése könyv leárazásakor azon felhasználóknak, akiknek szerepel a könyv a kívánságlistájukon.

```
CREATE OR REPLACE TRIGGER discount_on_wishlist
AFTER UPDATE OF discounted_price ON book
FOR EACH ROW
WHEN (OLD.discounted_price IS NULL)
DECLARE
    CURSOR wishlist_owners IS (SELECT w.customer_id
                                FROM wishlist w, partof p
                                WHERE w.wishlist_id = p.wishlist_id
                                AND p.book_id = :NEW.book_id
                                GROUP BY w.customer_id);
BEGIN
    FOR wishlist_owner IN wishlist_owners LOOP
        INSERT INTO notification (message, customer_id) VALUES
        (:NEW.title || ' című könyv a kívánságlistádon akciós lett!',
```

```

        wishlist_owner.customer_id);
    END LOOP;
END;
/

```

## Egyéb:

### Képernyőterv:

Megjegyzések: A szögletes zárójelek között szereplő szövegek a megjelenést befolyásoló tényezők, nem konkrét elemként értendők. A tervezet nem véglegesek, sokkal inkább távoli iránymutatók, biztosan át fognak esni módosításokon. Az [IFLI] jelölés az if logged in-et rövidíti, tehát azok a funkciók csak akkor érhetőek el, ha a felhasználó sikeresen bejelentkezett. A tervezett oldalak töredékére lett megvalósítva képernyőterv, viszont hasonló elképzélés alapján lesznek felépítve a további funkciók. A menüben a kép, melyben U betű szerepel valamelyen felhasználói ikonnal helyettesítendő.

Főoldal

The wireframe shows the main navigation bar with links: BookAround, Könyv (highlighted), Újdonság, Sikerlista, Áruházak, Kívánságlista [IFLI], Kosár [IFLI], and a user icon. Below the navigation, there's a section for 'New books' with placeholder text and three large placeholder boxes. A search bar with a placeholder 'Find the right book' and a 'Search' button follows. Under the search bar, a section titled 'Results (x items)' shows three book cards with placeholder data and a 'View more' button. At the bottom, there's a footer with links to 'Home' and 'Profile [IFLI]' and a copyright notice.

**Főoldal**

BookAround **Könyv** Újdonság Sikerlista Áruházak Kívánságlista [IFLI] Kosár [IFLI]

Legújabb könyveink...  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices.

Találja meg a megfelelő könyvet

[ Filterezési opciók ]

Találatok (x darab)

<b>Könyv címe</b> író(k) Ár <input type="button" value="Megtekintés"/>	<b>Könyv címe</b> író(k) Ár <input type="button" value="Megtekintés"/>	<b>Könyv címe</b> író(k) Ár <input type="button" value="Megtekintés"/>
---	---	---

[ további listázás ]

Főoldal | Profil [IFLI]  
BookAround Kft. minden jog fenntartva.

Könyv részletes adatainak listázása



BookAround

Könyv

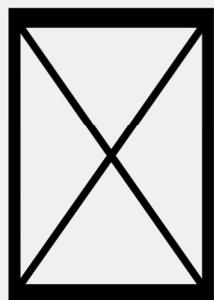
Újdonság

Sikerlista

Áruházak

Kívánságlista [IFLI]

Kosár [IFLI]



## Könyv címe

[ további összes adat ]

Eredeti ár:

Akciós ár:

[Kosárba](#)

[Kívánságlistára](#)

A következőket választották még



### Könyv címe

[ ... ]



### Könyv címe

[ ... ]



### Könyv címe

[ ... ]

[Főoldal](#) | [Profil \[IFLI\]](#)

BookAround Kft. minden jog fenntartva.

## Az áruházak leírása



BookAround

Könyv

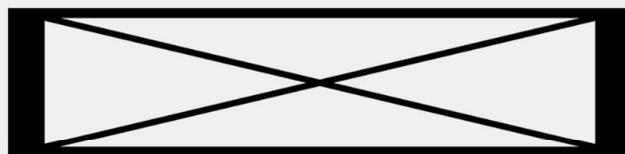
Újdonság

Sikerlista

Áruházak

Kívánságlista [IFLI]

Kosár [IFLI]



### BookAround - Tavasz pláza

[ áruház adatai ]

Nyitvatartás

Hétfő: 08:00 - 17:00

[ ... ]



[ ... ]

[Főoldal](#) | [Profil \[IFLI\]](#)

BookAround Kft. minden jog fenntartva.

## Felhasználói profil



BookAround

Könyv

Újdonság

Sikerlista

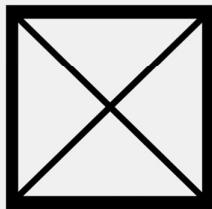
Áruházak

Kívánságlista [IFLI]

Kosár [IFLI]



### Üdvözlünk, Jónás!



Felhasználói adatok

E-mail cím

csakjonas@citromail.hu

[ További adatok listázása ]

csakjonas@citromail.hu

Adatok megváltoztatása

Kijelentkezés

Főoldal | Profil [IFLI]

BookAround Kft. minden jog fenntartva.

### CRUD táblázat:

Tábla	Create	Read	Update	Delete
Customer	X	X	X	X
Wishlist	X	X	X	X
Order	X	X	-	X
Invoicee	X	X	X (ki lehet fizetni)	-
Book	X	X	X	X
Author	X	X	X *	X
Genre	X	X	X *	X
Store	X	X	X	X
Stock	X	X	X	X
PartOf	X	X	-	X
Contains	X	X	-	X
Business_Hours	X	X	X	X
Notification	X	X	-	X

\* Az írók, műfajok létrehozása, frissítése és törlése a könyv módosításával lehetséges, tekintettel arra, hogy szorosan kötöttek egy könyvhöz. Módosítani nem lenne értelme, hiszen az egész tábla egy összetett kulcs, de gyakorlatilag módosíthatóak: egy könyvhöz tartozó írónak például átírhatjuk a nevét (az alkalmazás viszont törli a régit és létrehozza az újat a megfelelő adatokkal).

### Fordítás, futtatás:

JDK, Docker, Maven

## Adatbázis-szerver:

- Dockerrel futtatott (Oracle 12c)
  - a projekt root mappájában kiadott docker-compose up / docker compose up parancs felépíti
- Más módon futtatott adatbázis-szerver (Oracle 12c)
  - ebben az esetben a következő beállításokat várja el az alkalmazás:
    - `username: system`
    - `password: oracle`
    - `hostname: localhost`
    - `port: 1521`
    - `SID: xe`
- Ha fut a szerver, a system felhasználóval létrehozható a privát felhasználónk, melyre biztosított az `sql_scripts/dp_pl_sql.sql`-ben található PL/SQL script. Több funkciót is kezel, ebben a fázisban kizárolag a user opciót megadva létre fogja hozni a felhasználót a megfelelő adatokkal.
- Ha más hogy szeretnénk létrehozni, a következő beállításokat várja el az alkalmazás a felhasználónak:
  - `username: bookaround`
  - `password: bookaroundadmin`
- Ezek után váltsunk a bookstore felhasználóra. Itt létrehozhatjuk a táblákat, sequenceket, tárolt eljárásokat, függvényeket, triggereket. Mindkét létrehozó script működőképes, de az `sql_scripts/db_pl_sql.sql` több lehetőséget biztosít.
- A PL/SQL-es script használatával futtatjuk, majd a tables inputot adjuk. Létrejönnek a táblák, sequencek, tárolt eljárások, metódusok függvények. Ha hiba történik, azt remélhetőleg megfelelően jelzi. Újra futtatjuk a scriptet, ezúttal records inputtal, a megfelelő rekordokat beszűrja az adatbázisba.
- Ha újra szeretnénk inicializálni/létrehozni a táblákat és minden mást, az említett scriptnek a clean input adható, ez dropolja a táblákat, sequenceket (tehát a rekordokat is), a sequencek és tárolt eljárások/függvények nem kerülnek törlésre, viszont a következő létrehozásnál ez nem okoz problémát, ha léteznek akkor csak replaceli.

## Alkalmazás:

- Parancssori maven esetén
  - mvn spring-boot:run, ha minden igaz ez buildeli és futtatja az alkalmazást
- IntelliJ Idea esetén a projekt betöltése után használható a default konfiguráció vagy futtassuk a BookAroundApplication::main(String[] args) metódusát
- Ezek után az applikáció elérése a localhost:8080/index útvonalon történik (módosítható az application.properties-be).