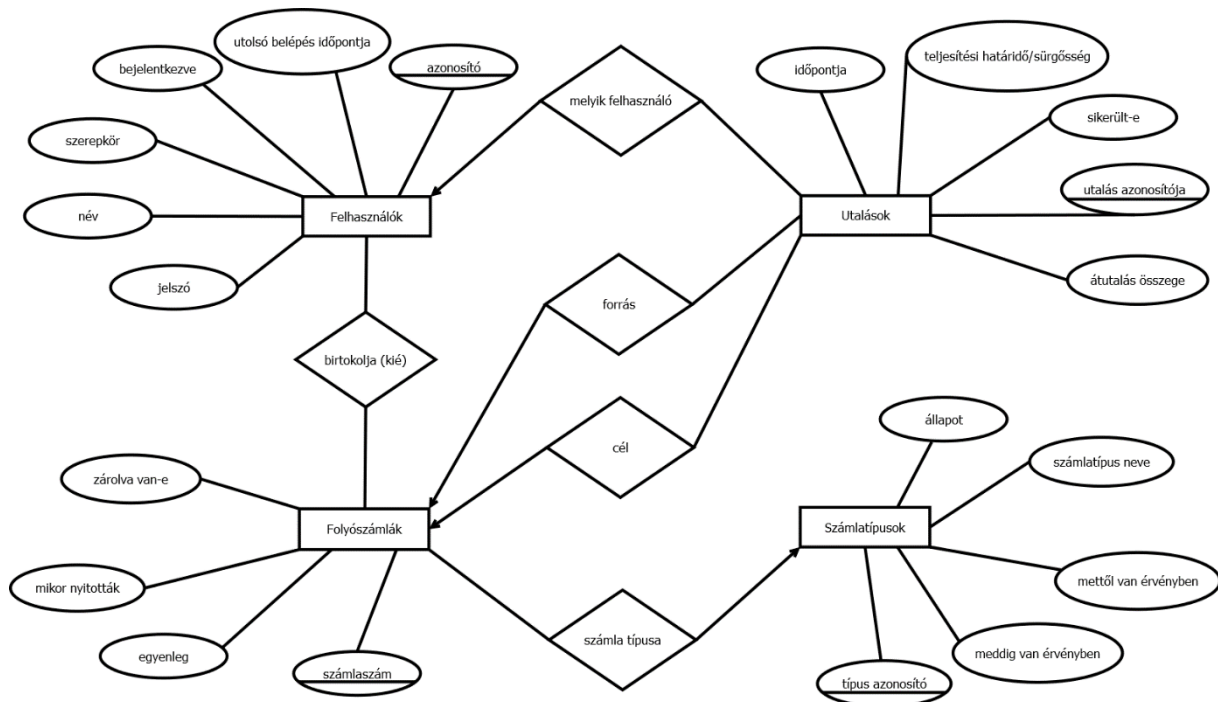


## Téma megnevezése

### Adatbázisok kötelező feladat

Egy banki adatbázisban nyilvántartjuk az ügyfeleket, a folyószámlákat és a pénzmozgásokat. A rendszerbe az ügyfelek és a banki alkalmazottak is be tudnak jelentkezni. Az ügyfelek és a banki alkalmazottak is tudnak utalásokat indítani, de az ügyfelek csak saját számláikról, a banki alkalmazottak viszont bármilyen számláról

### Egyed-kapcsolat modell



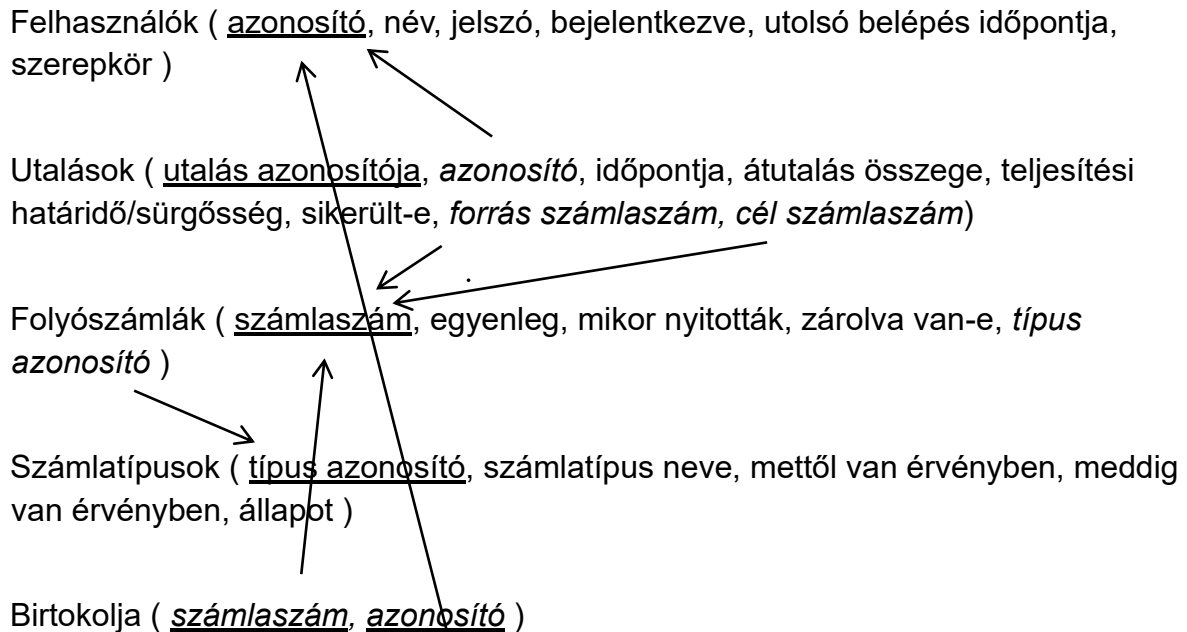
A téglalapokba írtak az egyedeket, a rombuszokba írtak a kapcsolatokat, míg az elipszisbe írtak az attribútumokat jelölik.

Az utalások és felhasználók közötti kapcsolat 1:N kapcsolat, hiszen egy felhasználó több utalást is indíthat, míg egy utalás csak egy felhasználóhoz tartozik. A felhasználók és folyószámlák közötti kapcsolat N:M kapcsolat, hiszen egy felhasználónak több folyószámlája is lehet, valamint egy folyószámla is tartozhat több

felhasználóhoz egyszerre. A folyószámlák és számlatípusok is 1:N kapcsolat, hiszen egy folyószámla csak egy típusú lehet, de több folyószámla is lehet azonos típusú. A folyószámlák és utalások szintűgy 2db 1:N kapcsolat, hiszen egy folyószámláról több utalás is indítható/fogadható, de egy utalás csak egy folyószámlához kapcsolható.

## Relációs adatbázisséma

---



[A külső kulcsokat dőlt betűvel jelöltem, míg az elsődleges kulcsokat aláhúzással. A külső kulcsokból a nyíl arra az attribútumra mutat, amelyekre hivatkozik.]

[Az 1:N kapcsolatokat a relációs sémává alakítás után be lehetett olvasztani az N oldali sémába. Ekkor a kapcsolat attribútumait valamint az 1 oldali séma kulcsát is hozzá írjuk az N oldali sémához]

[Az N:M kapcsolat nem olvasztható be egyik sémába sem. Ekkor új sémaként jelenik meg a saját attribútumával, valamint a kapcsolódó sémák kulcsaival.]

# Normalizálás

---

## 1NF:

Az összes relációsséma 1.NF-ben van, hiszen az attribútum-halmaz minden eleme atomi, tehát nincs benne többértékű és összetett attribútum sem.

## 2.NF:

A Felhasználók séma 2NF-ben van, mert minden másodlagos attribútum függ a kulcstól(azonosító).

A Utalások séma 2NF-ben van, mert minden másodlagos attribútum függ a kulcstól(utalás azonosítója).

A Folyószámlák séma 2NF-ben van, mert minden másodlagos attribútum függ a kulcstól(számlaszám).

A Számlatípusok séma 2NF-ben van, mert minden másodlagos attribútum függ a kulcstól(típus azonosító).

A Birtokolja séma 2NF-ben van, mert minden másodlagos attribútum függ a kulcstól({számlaszám, azonosító}).

## 3.NF:

A Felhasználók séma 3NF-ben van, mert nincs benne tranzitív függés.

Az Utalások séma 3NF-ben van, mert nincs benne tranzitív függés.

A Folyószámlák séma 3NF-ben van, mert nincs benne tranzitív függés.

A Számlatípusok séma 3NF-ben van, mert nincs benne tranzitív függés.

A Birtokolja séma 3NF-ben van, mert nincs benne tranzitív függés.

## Táblatervek

---

FELHASZNALOK		
azonosito	varchar(20)	A felhasználó azonosítója (kulcs)
nev	varchar(100)	A felhasználó neve
jelszo	varchar(100)	A felhasználó jelszava
bejelentkezve	boolean → tinyint(1)	A felhasználó be- és kijelentkezését tárolja
utolso_belepes_idopontja	timestamp	A felhasználó utolsó belépésének ideje
szerepkor	varchar(20)	A felhasználó szerepköre

FOLYOSZAMLAK		
szaamlaszam	int(11)	A folyószámla számlaszáma (kulcs)
egyenleg	decimal(20,0)	A folyószámla egyenlege
mikor_nyitottak	timestamp	A folyószámla nyitási ideje
zarolva_van	varchar(10)	A folyószámla állapota
tipus_azonosito	int(11)	A folyószámla típusa (külső kulcs)

SZAMLATIPUSOK		
tipus_azonosito	int(11)	A számlatípusok azonosítója (kulcs)
szamlatipus_neve	varchar(100)	A számlatípusok neve
mettol_van_ervenben	datetime	A számlatípusok érvényességének kezdete
meddig_van_ervenben	datetime	A számlatípusok érvényességének vége
allapot	varchar(10)	A számlatípusok állapota

UTALASOK		
utalas_azonositoja	int(11)	Az utalások azonosítója (kulcs)
azonosito	varchar(20)	Az utalást kezdeményező felhasználó azonosítója (külső kulcs)
idopontja	timestamp	Az utalások időpontja
atutalas_osszege	decimal(20,0)	Az utalások összege
teljesitesi_hatarido	datetime	Az utalások teljesítési határideje
sikerult	varchar(5)	Az utalások sikeressége
forras_szamlaszam	int(11)	Az utalást kezdeményező számlaszám (külső kulcs)
cel_szamlaszam	int(11)	Az utalás cél számlaszáma (külső kulcs)

BIRTOKOLJA		
azonosito	varchar(20)	A számla birtokosának azonosítója (külső kulcs)
szamlaszam	int(11)	A felhasználóhoz tartozó számlaszám (külső kulcs)

## Összetett lekérdezések

---

1. Ügyfelenként és havi bontásban (év, hónap) kilistázni az ügyfél által indított utalások összegét.  
(inditott.php-ben van megvalósítva)

```
SELECT YEAR(UTALASOK.teljesitesi_hatarido) AS ev,
MONTH(UTALASOK.teljesitesi_hatarido) AS honap, UTALASOK.azonosito,
SUM(UTALASOK.atutalas_osszege) AS osszeg, FELHASZNALOK.nev FROM
UTALASOK INNER JOIN FELHASZNALOK ON
FELHASZNALOK.azonosito=UTALASOK.azonosito GROUP BY ev, honap,
UTALASOK.azonosito, FELHASZNALOK.nev
```

2. Ügyfelenként és havi bontásban (év, hónap) kilistázni az ügyfél számláira érkező utalások összegét.  
(beerkezett.php-ben van megvalósítva)

```
SELECT FELHASZNALOK.azonosito AS id, FELHASZNALOK.nev AS nev,
YEAR(UTALASOK.idopontja) AS ev, MONTH(UTALASOK.idopontja) AS honap,
SUM(UTALASOK.atutalas_osszege) AS osszeg, UTALASOK.cel_szamlaszam
FROM UTALASOK INNER JOIN FOLYOSZAMLAJ ON
FOLYOSZAMLAJ.szamlaszam=UTALASOK.cel_szamlaszam INNER JOIN
BIRTOKOLJA ON FOLYOSZAMLAJ.szamlaszam=BIRTOKOLJA.szamlaszam
INNER JOIN FELHASZNALOK ON
FELHASZNALOK.azonosito=BIRTOKOLJA.azonosito WHERE
UTALASOK.sikerult='igen' GROUP BY ev, honap, id, nev
```

3. Kílistázni, hogy melyik ügyfélnek hány olyan folyószámlája van, amely a következő hónapban érvényét veszti.  
(ervenytelen.php-ben lett megvalósítva)

```
SELECT FELHASZNALOK.azonosito AS id, FELHASZNALOK.nev AS nev,  
COUNT(FOLYOSZAMLAK.szamlaszam) AS szamlak FROM FELHASZNALOK  
INNER JOIN BIRTOKOLJA ON BIRTOKOLJA.azonosito =  
FELHASZNALOK.azonosito INNER JOIN FOLYOSZAMLAK ON  
FOLYOSZAMLAK.szamlaszam = BIRTOKOLJA.szamlaszam INNER JOIN  
SZAMLATIPUSOK ON FOLYOSZAMLAK.tipus_azonosito =  
SZAMLATIPUSOK.tipus_azonosito WHERE  
MONTH(SZAMLATIPUSOK.meddig_van_ervenben) < MONTH(DATE_ADD(NOW(),  
INTERVAL 1 MONTH)) AND YEAR(SZAMLATIPUSOK.meddig_van_ervenben) =  
YEAR(DATE_ADD(NOW(), INTERVAL 1 MONTH)) GROUP BY id, nev
```

4. Kílistázni, hogy kiknek van olyan folyószámlájuk, ahol az egyenleg nulla.  
(egyenlegnelkuli.php-ban van megvalósítva)

```
SELECT DISTINCT FELHASZNALOK.azonosito, FELHASZNALOK.nev FROM  
FELHASZNALOK INNER JOIN BIRTOKOLJA ON  
BIRTOKOLJA.azonosito=FELHASZNALOK.azonosito INNER JOIN  
FOLYOSZAMLAK ON FOLYOSZAMLAK.szamlaszam=BIRTOKOLJA.szamlaszam  
WHERE FOLYOSZAMLAK.egyenleg=0
```

# Megvalósítás, funkciók

---

Az alkalmazást Notepad++-ban PHP és CSS nyelven írtam, az adatbázis létrehozásához pedig PHPMyAdmint használtam.

A megvalósított funkciók:

- A kezdőlapra a felhasználó ki tudja választani, hogy regisztrálni vagy már a meglévő fiókjába szeretne bejelentkezni. Ezt a megfelelő pontra kattintva tudja megtenni, ami átirányítja az adott oldalra.
- A regisztrációs oldalon a felhasználó a kért adatokat megadva majd a regisztráció gombra kattintva tudja létrehozni a fiókját. Ekkor a rendszer átirányítja a bejelentkezés oldalra és visszajelez a regisztráció sikerességéről. Ha bármelyik adatot hibásan adja meg a felhasználó arról is kap visszajelzést.
- A bejelentkezés oldalon a felhasználó az oldal által kért saját adatait megadva majd a bejelentkezés gombra kattintva tud belépni a fiókjába. Ekkor a rendszer átirányítja a bejelentkezett kezdőlapra és a bejelentkezés sikerességéről is kap egy visszajelzést. Ha bármelyik adatát nem jól adja meg, arról is kap visszajelzést.
- Belépéskor más "kezdő" oldal fogadja az ügyfelet és az alkalmazottat. Mindketten a fenti menüből tudnak választani, attól függően mit szeretnének tenni.

Ügyfél:

- A számláim oldalon az ügyfél tud új számlát nyitni, a meglévő számláját törölni, befizetni rá, nyitni/zárni, valamint az általa kiválasztott személyekkel megosztani. A felhasználó az oldal alján található táblázat segítségével tudja nyomon követni számlái állapotát. Bármelyik adatot hibásan adja meg visszajelez a rendszer, valamint a sikerességről is ad visszajelzést.
- Az utalások oldalon tud az ügyfél a saját számláiról a kiválasztott számlára a kívánt összeget átutalni. Itt is az oldal alján található táblázat segítségével tudja nyomon követni az utalásait, annak sikerességét! Bármelyik adatot hibásan adja meg visszajelez a rendszer, valamint a sikerességről is ad visszajelzést.

Alkalmazott:

- A számlák kezelése oldalon az alkalmazott tud új számlát nyitni, a meglévő számláját törölni, befizetni rá, nyitni/zárni, valamint az általa kiválasztott személyekkel megosztani. Az alkalmazott az oldal alján található táblázat segítségével tudja nyomon követni az összes felhasználó számláinak



állapotát. Bármelyik adatot hibásan adja meg visszajelez a rendszer, valamint a sikerességről is ad visszajelzést.

- Az utalások kezelése oldalon tud az alkalmazott az összes felhasználó számlájáról a kiválasztott számlára a kívánt összeget átutalni. Itt is az oldal alján található táblázat segítségével tudja nyomon követni az összes kezdeményezett utalást, annak sikerességét! Bármelyik adatot hibásan adja meg visszajelez a rendszer, valamint a sikerességről is ad visszajelzést.
- A számlatípusok létrehozása oldalon az alkalmazott tud új számlatípust létrehozni, valamint a meglévő típust passzíváltatni. Az oldal alján található táblázat segítségével tudja nyomon követni a számlatípusok állapotát. Bármelyik adatot hibásan adja meg visszajelez a rendszer, valamint a sikerességről is ad visszajelzést.
- Az indított utalások összege, beérkezett utalások összege, érvénytelen számlák és az egyenleg nélküli számlák oldalon a lekérdezések eredményei tekinthetők meg.

Mindkét felhasználó esetében:

- A vissza gombra kattintva bármelyik oldalról vissza tud lépni a felhasználó a bejelentkezés utáni kezdőlapra.
- A kijelentkezés gombra kattintva pedig kijelentkezni tud a felhasználó, ekkor a bejelentkezés oldalra irányítja vissza az alkalmazás, aminek sikerességéről visszajelzést is ad.