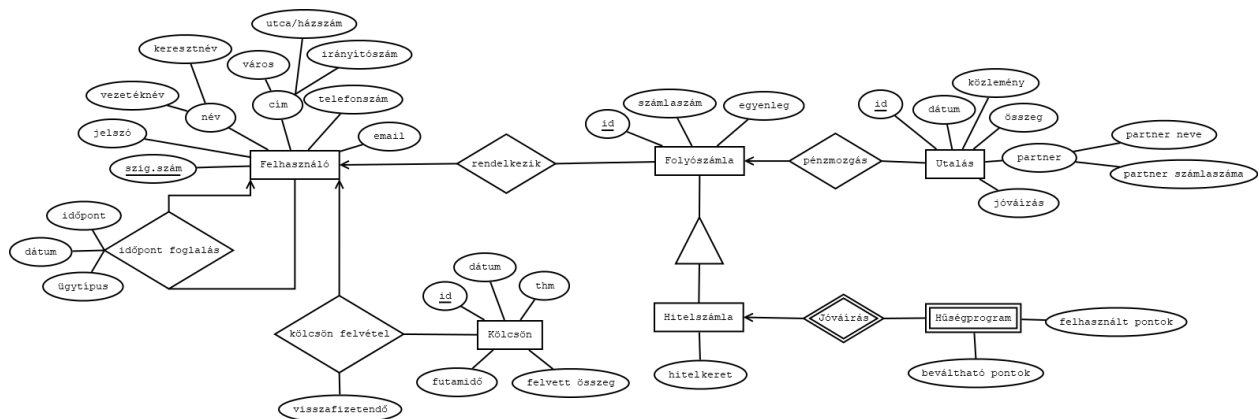


Bank

Adatbázisok kötelező feladat

A program feladata, hogy az alapvető pénzügyi mozgásokat egyszerűen nyomon követhetővé tegye, egy letisztult, csak a lényegre koncentráló adatkezelési felülettel. A felhasználónak lehetősége van utalás kezdeményezésre, személyes ügyintézéshez időpont foglalásra, kölcsön igénylésre, illetve teljes vagy részleges számlatörténet lekérdezésére. A program használatához az ügyfélnek a banknál személyesen kell igényelnie saját felhasználói fiókot.

Egyed-kapcsolat modell



A felhasználó egyed lesz, különböző tulajdonságokkal jellemezhetjük. A személyi igazolvány szám biztosan kulcs lesz, hiszen ez egyértelműen azonosítja a felhasználót. Egy felhasználónak van neve, címe, telefonszáma, e-mail címe és jelszava, amit a bank, regisztráláskor ad meg. A folyószámla szintén egyed lesz, minden folyószámlának lesz egy

egyedi azonosítója, továbbá rendelkezni fog számlaszámmal és egyenleggel. Az utalás egyed rendelkezik egy id-val, amivel egyszerűen be lehet majd azonosítani. Kell lennie dátumának, hogy mely napon történt az utalás. Minden pénzmozgáshoz írhatunk közleményt, illetve meg kell, hogy adjuk az utalandó összeget és azt, hogy kinek kívánjuk küldeni, ezek lesznek a további attribútumai az egyednek. A hitelszámla egy altípus, hitelkeret attribútummal. Az EK-diagrammban továbbá található még egy gyenge egyed, hűségprogram néven, melynek két attribútuma van, a beváltható pontok és a már felhasznált pontok. A hitelszámláról való utalás során a beváltható pontok gyarapodnak, melyeket a későbbiekben fel is használhatunk. A kölcsön is egy egyed lesz, mely kap egy id-t a könnyeb beazonosításhoz, egy dátumot, mely napon történt a hitel felvétele, egy teljes hiteldíj mutatót, illetve a futamidőt, amik segítségével a felvett összegből könnyen ki lehet számolni a teljes, visszafizetendő összeget.

Az időpont foglalásakor, egy felhasználó több különböző időpontot is felvehet, viszont egy bizonyos időpontot csak egy felhasználó vehet fel. 1:N típusú kapcsolathoz jutunk, a kapcsolat 3 attribútummal rendelkezik, időpont, ügytípus és dátum. A felhasználó rendelkezhet számlákkal, tehát egy felhasználónak több folyószámlája is lehet, azonban egy számlának csak egy tulajdonosa lehet. 1:N kapcsolat van közöttük. A felhasználó kölcsönt vehet fel, akár többet is, a kölcsönök csak egy felhasználóhoz tartozhatnak. Ebben az esetben is 1:N kapcsolatról beszélhetünk. A folyószámla, mint főtípus kapcsolatát az altípussal specializáló kapcsolattal adhatjuk meg, a csúcsával a főtípus felé mutató háromszöggel. Mivel a hűségprogram gyenge egyedként szerepel, szükségünk van egy meghatározó kapcsolatra, mivel ezeket az egyedeket is meg kell tudnunk határozni. Utalás közben pénzmozgás történik a számláinkon, egy utalás csak egy folyószámláról indítható, azonban egy számláról akár több utalást is kezdeményezhetünk, így itt is 1:N kapcsolat jön létre.

Relációs adatbázisséma

FELHASZNÁLÓ (szig.szám, jelszó, vezetéknév, keresztnév, irányítószám, város, utca/házszám, telefonszám, email)

IDŐPONT FOGLALÁS (időpont, dátum, ügytípus, FELHASZNÁLÓ.szig.szám)

FOLYÓSZÁMLA (id, számlaszám, egyenleg, hitelszámla, hitelkeret, FELHASZNÁLÓ.szig.szám)

HŰSÉGPROGRAM (beváltható pontok, felhasznált pontok, FOLYÓSZÁMLA.id)

UTALÁS (id, dátum, közlemény, összeg, partner neve, partner számlaszáma, FOLYÓSZÁMLA.id, jóváírás)

KÖLCSÖN (id, dátum, thm, futamidő, felvett összeg, FELHASZNÁLÓ.szig.szám, visszafizetendő)

Normalizálás

A relációs adatbázisséma 1NF-ben van, mert nincs összetett attribútuma.

FELHASZNÁLÓ (szig.szám, jelszó, vezetéknév, keresztnév, irányítószám, város, utca/házszám, telefonszám, email)

2NF: A FELHASZNÁLÓ séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulcstól.

3NF: A FELHASZNÁLÓ séma nincs 3NF-ben mert az irányítószámtól függ a város.

FELHASZNÁLÓ (szig.szám, jelszó, vezetéknév, keresztnév, irányítószám, utca/házszám, telefonszám, email)

VÁROS (irányítószám, város)

IDŐPONT FOGLALÁS (időpont, dátum, ügypont FELHASZNÁLÓ.szig.szám)

2NF: A IDŐPONT FOGLALÁS séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulcstól.

3NF: A IDŐPONT FOGLALÁS séma 3NF-ben van, mert nem tudunk függést kijelölni.

FOLYÓSZÁMLA (id, számlaszám, egyenleg, hitelkeret)

2NF: A FOLYÓSZÁMLA séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulcstól.

A FOLYÓSZÁMLA séma 3NF-ben van, mert nem tudunk függést kijelölni.

HŰSÉGPROGRAM (beváltható pontok, felhasznált pontok, FOLYÓSZÁMLA.id)

2NF: A HŰSÉGPROGRAM séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulcstól.

3NF: A HŰSÉGPROGRAM séma 3NF-ben van, mert nem tudunk függést kijelölni.

UTALÁS (id, dátum, közlemény, összeg, partner neve, partner számlaszáma, jóváírás, *FOLYÓSZÁMLA.id*)

2NF: A UTALÁS séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulestől.

3NF: A UTALÁS séma nincs 3NF-ben mert a partner számlaszámától függ a partner neve.

UTALÁS (id, dátum, közlemény, összeg, *partner számlaszáma*, jóváírás, *FOLYÓSZÁMLA.id*)

PARTNER (partner számlaszáma, partner neve)

A UTALÁS séma 3NF-ben van, mert nem tudunk több függést kijelölni.

KÖLCSÖN (id, dátum, thm, futamidő, felvett összeg, visszafizetendő, *FELHASZNÁLÓ.szig.szám*)

2NF: A KÖLCSÖN séma 2NF-ben van, mert a kulcsattribútumok halmaza egyetlen elemből áll, ezért minden másodlagos attribútum teljesen függ a kulestől.

FELHASZNÁLÓ (szig.szám, jelszó, vezetéknev, keresztnév, *VÁROS.irányítószám*, utca/házszám, telefonszám, email)

VÁROS (irányítószám, város)

FOLYÓSZÁMLA (id, számlaszám, egyenleg, hitelkeret, *FELHASZNÁLÓ.szig.szám*)

UTALÁS (id, dátum, közlemény, összeg, *partner számlaszáma*, jóváírás, *FOLYÓSZÁMLA.id*)

PARTNER (partner számlaszáma, partner neve)

KÖLCSÖN (id, dátum, thm, futamidő, felvett összeg, visszafizetendő, *FELHASZNÁLÓ.szig.szám*)

HŰSÉGPROGRAM (beváltható pontok, felhasznált pontok, *FOLYÓSZÁMLA.id*)

IDŐPONT FOGLALÁS (időpont, dátum, *ÜGYTIPUS.id*, *FELHASZNÁLÓ.szig.szám*)

ÜGYTIPUS (id, ügytípus megnevezés)

Táblatervek

bank felhasznalo # szig_szam : varchar(8) # jelszo : varchar(100) # vezeteknev : varchar(50) # keresztnév : varchar(50) # irányítószám : int(4) # utca/házszám : varchar(50) # email : varchar(100) # telefonszám : varchar(11) # profilkep : text	bank folyoszamla # id : int(6) # számlaszám : varchar(26) # egyenleg : int(20) # felhasznalo : varchar(8) # hitelszámla : int(11) # hitelkeret : int(11)	bank utalas # id : int(11) # partner_számlaszám : varchar(26) # datum : date # közlemény : varchar(50) # összeg : int(20) # folyoszamla : int(11) # jóváírás : int(1)
bank kölcsön # id : int(11) # datum : date # thm : float # futamido : int(3) # felvett_összeg : int(20) # felhasznalo : varchar(8) # visszafizetendo : int(11)	bank husegprogram # bevalthato_pontok : int(11) # felhasznalt_pontok : int(11) # folyoszamla_id : int(6)	bank partner # partner_számlaszám : varchar(26) # partner_neve : varchar(30)
	bank idopont_foglalas # idopont : varchar(5) # datum : date # ugytipus : int(3) # felhasznalo : varchar(8)	bank ugytipus # id : int(11) # ugytipus_megnevezes : varchar(100)
		bank varos # irányítószám : int(4) # varos : varchar(11)

Összetett lekérdezések

```
SELECT SUM(`visszafizetendo`/`futamido`) as `osszeg`,(SELECT `szamlaszam` FROM  
`folyoszamla` WHERE `felhasznalo` = "'.$_SESSION["aktiv"].'" AND `egyenleg` >  
0) as `szamla` FROM `kolcson` WHERE `felhasznalo` = "'.$_SESSION["aktiv"].'"  
group BY `szamla`'
```

A fenti SQL lekérdezéssel összegezzük a felvett hitelek havi törlesztőjét, az allekérdezésben csak azt a folyószámlát kérem le, amelyik egyenlege nagyobb 0-nál, mivel így az biztosan nem hitelszámla lesz. A SUM függvény összeadja a visszafizetendő összeg havi bontása utáni összegeket. A lekérdezés végén pedig csoportosítja azokat számlaszám alapján.

```
"SELECT `partner`.`partner_neve` ,`utalas`.`partner_szamlaszam`, SUM(`osszeg`)
AS `osszeg` FROM `utalas` INNER JOIN `partner` ON
`utalas`.`partner_szamlaszam` = `partner`.`partner_szamlaszam` WHERE
`folyoszamla` = " . $_SESSION["id"] . " AND `datum` BETWEEN '" . $kezdet . "'
AND '" . $veg . "' GROUP BY `partner`.`partner_szamlaszam`";
```

Ebben a lekérdezésben kérjük le a partner nevét, illetve számlaszámát. Tulajdonképpen egy összesített lekérdezést szeretnénk készíteni, egy partner részére elutalt teljes összegre a kiválasztott dátumok között. Két táblát kapcsolunk össze, az utalas és a partner táblákat, az utalas táblában lévő partner_szamlaszam és a partner táblában lévő partner_szamlaszam mentén. Azok a számlaszámok kerülnek lekérésre, amik a SESSION által eltárolt azonosítóval vannak kapcsolatban. A lekérdezés végén pedig a GROUP BY segítségével csoportosítom a partnereket.

Az alábbi lekérdezés segítségével tudjuk a számlatörténet lekérésénél kiírni az egyenleget. Két belső SELECT utasítással lekérjük a számlára beérkezett és a számláról kimenő összegeket, majd attól függően, hogy utalás vagy bevétel volt, kivonjuk vagy hozzáadjuk az aktuális egyenleghez. Erre azért van szükség, mert a felhasználó számlatörténet lekérdezésekor tetszőleges dátumokat választhat ki, így biztosítva lesz az akkori aktuális egyenleg. Azonban előfordulhat olyan hónap melyben nem történt utalás vagy nem érkezett bevétel, ezért az IFNULL funkcióval ezeket a NULL értékeket 0-ra változtatjuk. Az így kapott egyenleget GROUP BY segítségével összegzem.

```
"SELECT (`egyenleg`-IFNULL((SELECT SUM(`osszeg`) FROM `utalas` WHERE
`folyoszamla` = " . $_SESSION["id"] . " AND `jovairas` = 0 AND `datum` BETWEEN
'" . $veg . "' AND '".date("Y-m-d")."',0) + IFNULL((SELECT SUM(`osszeg`) FROM
`utalas` WHERE `folyoszamla` = " . $_SESSION["id"] . " AND `jovairas` = 1 AND
`datum` BETWEEN '" . $veg . "' AND '".date("Y-m-d")."',0)) as egyenleg FROM
`folyoszamla` INNER JOIN `utalas` ON `folyoszamla`.`id` =
`utalas`.`folyoszamla` WHERE `folyoszamla` = " . $_SESSION["id"] . " AND
`datum` BETWEEN '" . $veg . "' AND '".date("Y-m-d")."' GROUP BY `egyenleg`"
```

Megvalósítás

A projekt a következő technológiákat/szoftvereket fogja használni a futása során:

- PhpMyAdmin, webes felületű kliens az XAMPP-on belül MySQL adatbázis kezeléséhez
- PHP(backend)
- CSS(frontend)
- Bootstrap(frontend)
- Ajax
- JavaScript(frontend)
- HTML

Az elkészült alkalmazás funkciói

A programot bejelentkezés után lehet használni, az ehhez és regisztráláshoz szükséges adatokat a bank biztosítja számunkra. Az egyszerűség kedvéért mind a felhasználónév mind a jelszó most **teszt**. Az ügyfél felhasználó névként a személyi igazolvány számára lesz szüksége, jelszónak pedig a bank által generált kódot használhatja.

Bejelentkezés után választhatjuk ki, hogy mely folyószámlánkról szeretnénk utalni vagy esetleg ügyet intézni. A kiválaszt gombra kattintva a fő oldalra irányít a program, ahol több opció közül is választhatunk.

Egerünket egy-egy menüpont fölé húzva, további opciók jelennek meg. Számlaváltás a jobb felső sarokban a navigációs sávon található „számla váltás” gombra kattintva lehetséges.

- Utalások
 - utalás: a kiválasztott számlánkról utalhatunk, a szükséges adatok megadása után
 - korábbi utalások: utolsó 8 utalásunk
 - számlatörténet lekérése: kezdő és záró dátum választás után a program listázza az utalási adatokat, itt lehet választani részletes, összesített és diagram szerinti

lekérdezés között, nyomtatás gombra kattintva egy letisztultabb, táblázatos kimutatást menthetünk le vagy nyomtathatunk ki

- Időpontok
 - időpont foglalás: időpont foglaláshoz szükséges kiválasztani a dátumot, időpontot illetve az ügýtípust, egy felhasználó egy időpontot csak egyszer foglalhat le, erre a program is figyelmeztet
 - lefoglalt időpontjaim: az eddig lefoglalt időpontok, törölni visszamenőleg nincs lehetőség
- Kölcsönök
 - kölcsöneim: az eddig felvett kölcsönökről egy lista, amely magába foglalja a fontosabb adatokat, illetve egy összesítés, amelyből láthatjuk, hogy havi szinten mennyit von a bank a folyószámlánkról
 - új kölcsön igénylése: szükséges adatok megadása után, 500.000Ft alatti havi nettó jövedelem mellett 50%-ig, míg 500.000Ft felett a jövedelem 60%-ig terhelhető le a felhasználó jövedelme.
- Beállítások
 - saját adatok módosítása: a felhasználó választhat saját profilképet magának, a csillaggal jelölt adatok módosíthatók, ha a felhasználó szeretne változtatni a többi adatán, akkor be kell fáradnia egy bankfiókba személyesen
 - hűségpontjaim: hitelszámláról való utalás után, az elutalt összeg 1%-át a felhasználó megkapja úgynevezett hűségpontok formájában, melyet beválthat a felsorolt lehetőségek egyikére
 - partner lista: partnerek táblázatos megjelenítése, akiknek a felhasználó már utalt, vagy akiktől már folyt be összeg a számlájára