JEGYZŐKÖNYV

Adatbázisrendszerek I.

# Féléves feladat

Könyvtári nyilvántartás létrehozása

Készítette: **Bartók-Balog Péter**

Neptunkód: **ASNQPH**

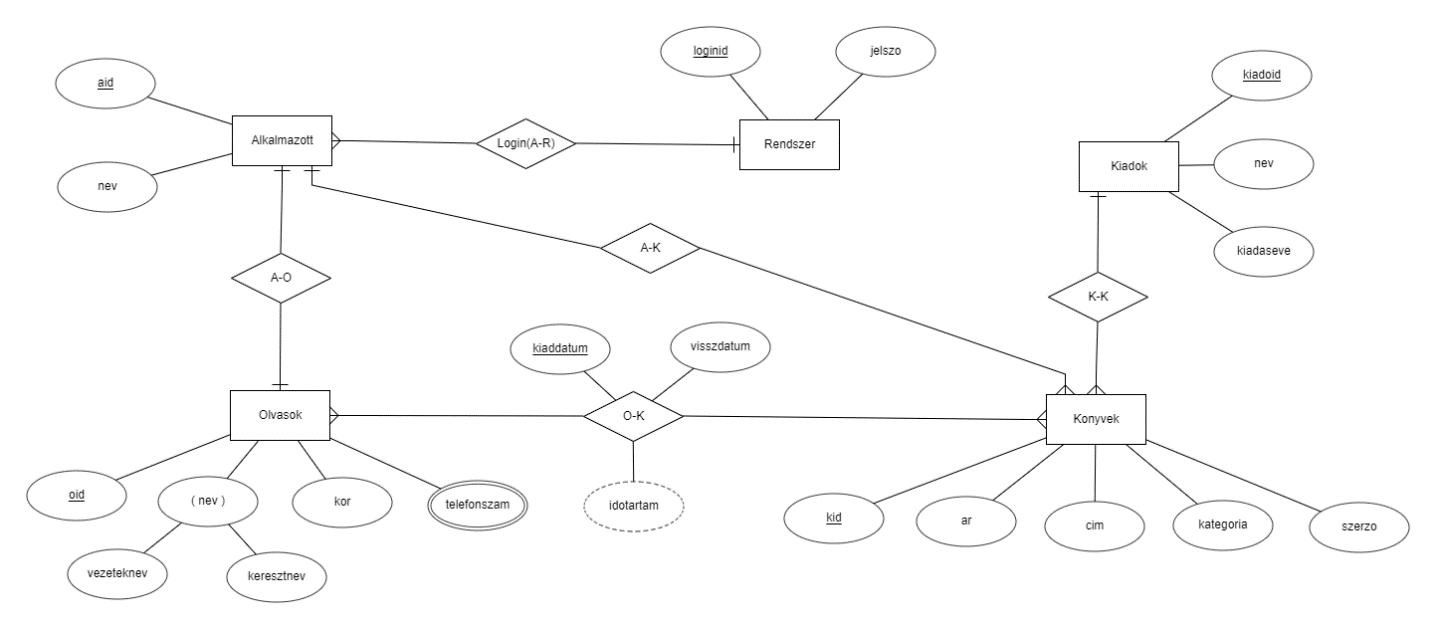
**A feladat leírása:**

A feladat egy hiteles adatbázis létrehozása, különböző lekérdezések kialakítása mysql nyelven keresztül. A feladat megoldásának feltétele, hogy az adatbázis tartalmazzon legalább 5 egyedet, egyedenként 4 tulajdonságot, valamint többféle kapcsolatot. A probléma megoldása kezdődik az adatbázis témájának kiválasztásával, ami jelen esetben egy könyvtári rendszer adatbázisa. Ezt követi az adatbázis szerkezetét leíró ER, valamint Relációs modelljének kialakítása. Amint a megfelelő konvertálások megtörténetek a feladat folytatódik a táblák létrehozásával és feltöltésével. Miután a táblák feltöltődtek adatokkal, létre kell hozni 5 egyszerű, és 5 bonyolult lekérdezést, amelyek megoldását a relációs algebra nyelvén is meg kell adni. A feladat befejezéshez szükséges a létrehozott dokumentum, illetve adatbázis megvédése a határidőben feltüntetett gyakorlat időpontjáig.

**ER modell leírása:**

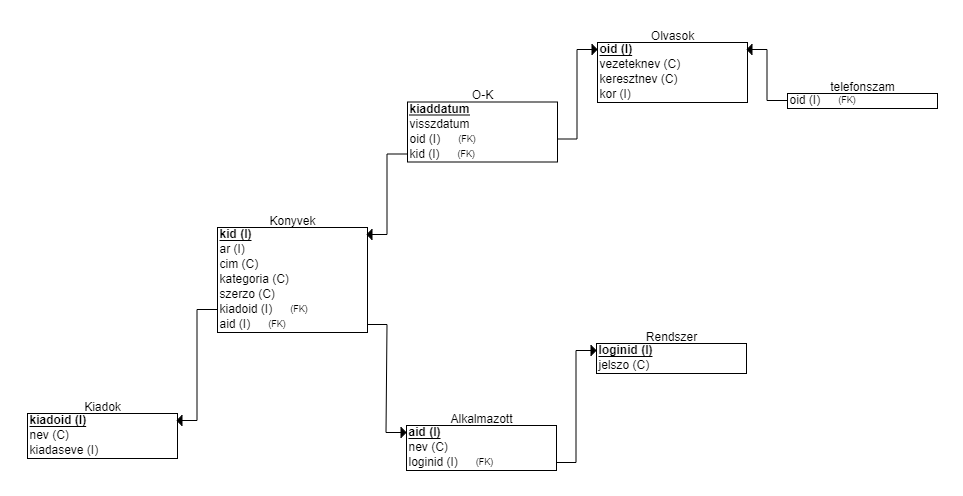
A feladatban egy reális, és valós E-R adatbázist hozok létre. Az adatbázisnak a következőek a jellemzői:

* Alkalmazott egyed: „**nev**” **egyedi tulajdonság**, „**aid**” tulajdonság.
* Rendszer egyed: „**loginid**” **egyedi tulajdonság** és „**jelszo**”, azaz a belépéshez szükséges adatok.
* Olvasok egyed: „**oid**” **egyedi tulajdonság**, „**nev**” **összetett tulajdonság** („vezeteknev”, „keresztnev”), „**kor**”, „**telefonszam**” **többértékű tulajdonság**, hiszen többet telefonszám is lehetséges.
* Konyvek egyed: „**kid**” **egyedi tulajdonság**, „**ar**”, „**cím**”, „**kategoria**”, „**szerzo**” tulajdonságok magától értetődőek.
* Kiadok egyed: „**kiadoid**” **egyedi tulajdonság**, „**nev**”, „**kiadaseve**” magától értetődő.
* O-K kapcsolat: **N-M** kapcsolat „**idotartam**” tulajdonsága **származtatott**, „kiaddatum” és „visszdatum” hányadosa adja.
* K-K kapcsolat: **1-N** kapcsolat „Konyvek”, és „Kiadok” között, hiszen egy kiadó több könyvet is kiadhat.
* A-K kapcsolat: **1-N** kapcsolat. Az alkalmazottak több könyvet is ápolhatnak.
* A-R kapcsolat: **1-N** kapcsolat. A „Rendszer” egyed több alkalmazottnál is előfordulhat.



**Adatbázis konvertálása relációs modellre:**

* „Alkalmazott” egyedből egy tábla lesz, a meglévő tulajdonságokon kívül lesz még, egy „loginid” idegen kulcs is. „aid” mező automatikusan növekszik.
* „Olvasok” egyedből egy tábla lesz, tulajdonságai közül a „telefonszam” tulajdonság egy külön táblába kerül, amely tartalmazza „oid” elsődleges kulcsot „Olvasok” táblából. A „nev” összetett tulajdonság kimarad, helyére „keresztnev”, és „vezeteknev” tulajdonságok kerülnek. „oid” automatikusan növekszik, „kor” érteke minimum 16 lehet.
* „Kiadok” egyedből egy tábla lesz, tulajdonságai maradnak ugyan azok. „kiadoid” automatikusan növekszik, „nev” mező nem lehet üres, „kiadaseve” nem lehet nulla.
* „Rendszer” egyedből szintén egy tábla lesz, tulajdonságai maradnak. „loginid”, és „jelszo” mezők nem lehetnek nullák vagy üresek.
* „Konyvek” egyedből egy tábla lesz meglévő tulajdonságain kívül három idegenkulcs tartja a kapcsolatot „Olvasok”, „Kiadok”, és „Alkalmazott” tábla között. „kid” automatikusan növekszik, „cim”, „kategoria”, „szerzo”, nem lehet üres.
* O-K kapcsolat N-M kapcsolat ezért egy külön táblába kerül. Meglévő tulajdonságokon kívül két idegenkulcs tartja a kapcsolatot „Konyvek”, és „Olvasók” között. „idotartam” tulajdonság származtatott, így az kimarad a táblából.



**Adatbázis relációs sémája:**

* Alkalmazott [aid, nev, loginid]
* Rendszer [loginid, jelszo]
* O-K [kiaddatum, visszdatum, kid, oid]
* Olvasok [oid, vezeteknev, keresztnev, kor]
* Telefonszam [oid]
* Konyvek [kid, ar, cím, kategoria, szerzo, oid, kiadoid, aid]
* Kiadok [kiadoid, nev, kiadaseve]

**A táblák létrehozása:**

A táblák létrehozásánál ügyeltem azoknak sorrendjére, hogy az idegen kulcsok beillesztéséhez szükséges elsődleges kulcs már rendelkezésre álljon a létrehozandó táblánál. Szintén fontos, hogy az elsődleges kulcs típusa megegyezzen az idegen kulcséval, ami az én esetemben csak INT típus.

* CREATE TABLE OLVASOK (OID INT PRIMARY KEY, VEZETEKNEV CHAR(50) NOT NULL, KERESZTNEV CHAR(20) NOT NULL, KOR INT CHECK (KOR>12));
* CREATE TABLE RENDSZER (LOGINID INT PRIMARY KEY, JELSZO CHAR (50) NOT NULL);
* CREATE TABLE ALKALMAZOTT (AID INT PRIMARY KEY, NEV CHAR (50) NOT NULL, LOGINID INT, FOREIGN KEY (LOGINID) REFERENCES RENDSZER(LOGINID));
* CREATE TABLE KIADOK (KIADOID INT PRIMARY KEY, NEV CHAR(50) NOT NULL, KIADASEVE INT NOT NULL);
* CREATE TABLE KONYVEK (KID INT PRIMARY KEY, AR INT NOT NULL, CIM CHAR (50) NOT NULL, KATEGORIA CHAR (20) NOT NULL, SZERZO CHAR (50) NOT NULL, OID INT, FOREIGN KEY (OID) REFERENCES OLVASOK(OID), KIADOID INT, FOREIGN KEY (KIADOID) REFERENCES KIADOK(KIADOID), AID INT, FOREIGN KEY (AID) REFERENCES ALKALMAZOTT(AID));
* CREATE TABLE TELEFONSZAM (OID INT, FOREIGN KEY(OID) REFERENCES OLVASOK(OID));
* CREATE TABLE OK (KIADDATUM DATE NOT NULL, VISSZDATUM DATE, KID INT, FOREIGN KEY(KID) REFERENCES KONYVEK(KID), OID INT, FOREIGN KEY(OID) REFERENCES OLVASOK(OID));

**A táblák feltöltése:**

RENDSZER TÁBLA:

INSERT INTO RENDSZER VALUES(1,'VALAMI');

INSERT INTO RENDSZER VALUES(2,'VALAMI1');

INSERT INTO RENDSZER VALUES(3,'VALAMI2');

INSERT INTO RENDSZER VALUES(4,'VALAMI3');

INSERT INTO RENDSZER VALUES(5,'VALAMI4');

ALKALMAZOTT TÁBLA**:**

INSERT INTO ALKALMAZOTT VALUES(1,'KOVÁCS ISTVÁN',1);

INSERT INTO ALKALMAZOTT VALUES(2,'BALÁZS LÁSZLÓ',2);

INSERT INTO ALKALMAZOTT VALUES(3,'ISTVÁN GÁBOR',3);

INSERT INTO ALKALMAZOTT VALUES(4,'SZABÓ GÁBOR',4);

INSERT INTO ALKALMAZOTT VALUES(5,'RUSZÓ ERIK',5);

KIADOK TÁBLA:

INSERT INTO KIADOK VALUES(1,'PARAMOUNT',2011);

INSERT INTO KIADOK VALUES(2,'JÓ KIADÓ',2012);

INSERT INTO KIADOK VALUES(3,'ROSSZ KIADÓ',2013);

INSERT INTO KIADOK VALUES(4,'LIBRI',2014);

INSERT INTO KIADOK VALUES(5,'MÓRA',2015);

OLVASOK TABLA:

INSERT INTO OLVASOK VALUES(1,'TAKÁCS','ERNŐ',18);

INSERT INTO OLVASOK VALUES(2,'BÍRÓ','ERNŐ',19);

INSERT INTO OLVASOK VALUES(3,'DEÁK','FERENC',20);

INSERT INTO OLVASOK VALUES(4,'TÓTH','PÉTER',30);

INSERT INTO OLVASOK VALUES(5,'LUKÁCS','BENCE',35);

KONYVEK TABLA:

INSERT INTO KONYVEK VALUES(1,1000,'A KÖNYV','DRÁMA','ARANY JÁNOS',1,1,1);

INSERT INTO KONYVEK VALUES(2,2000,'A MÁSIK KÖNYV','LÍRA',NULL,2,2,2);

INSERT INTO KONYVEK VALUES(3,3000,'HARRY POTTER','MISZTIKUS','JK ROWLING',3,3,3);

INSERT INTO KONYVEK VALUES(4,2500,'LEGENDÁS ÁLLATOK','MISZTIKUS','JK ROWLING',4,4,4);

INSERT INTO KONYVEK VALUES(5,5000,'MAKK MARCI','KALAND','ÁRPÁD BÉLA',5,5,5);

OK TÁBLA:

INSERT INTO OK VALUES('2021-05-04','2021-06-01',1,1);

INSERT INTO OK VALUES('2021-04-20','2021,04-28',2,2);

INSERT INTO OK VALUES('2021-07-02','2021-07-18',3,3);

INSERT INTO OK VALUES('2021-08-01','2021-09-02',4,4);

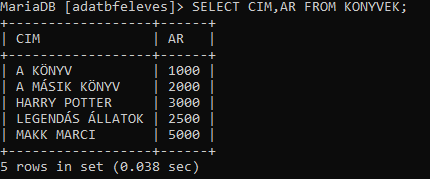
INSERT INTO OK VALUES('2021-01-05',NULL,5,5);

**Egyszerűbb lekérdezések:**

1. Készítsünk lekérdezést, amely kiadja az egyes könyvek árát! (cim,ar)

**SELECT CIM, AR FROM KONYVEK;**

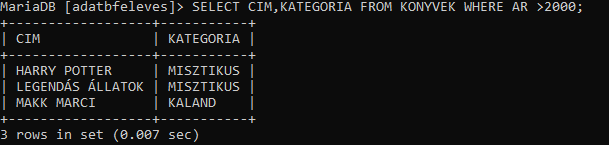
**πcim, arkonyvek**



1. Készítsünk lekérdezést, amely megadja a 2000-nél drágább könyvek címét, és kategóriáját! (cim,kategoria)

**SELECT CIM, KATEGORIA FROM KONYVEK WHERE AR >2000;**

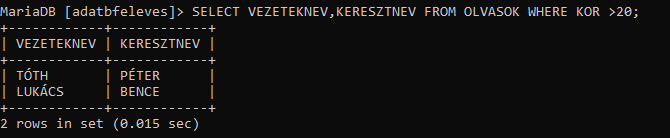
**πcim, kategoria σar > 2000konyvek**



1. Készítsünk lekérdezést, amely megadja a 20 évnél idősebb olvasók nevét! (vezeteknev,keresztnev)

**SELECT VEZETEKNEV, KERESZTNEV FROM OLVASOK WHERE KOR >20;**

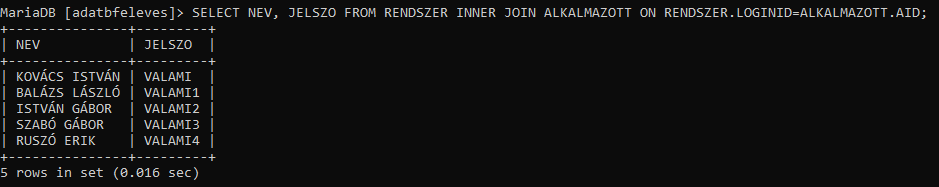
**πvezeteknev, keresztnev σkor > 20olvasok;**



1. Készítsünk lekérdezést, amely megadja az alkalmazottak nevét, és a belépéshez szükséges jelszót! (nev,jelszo)

**SELECT NEV, JELSZO FROM RENDSZER INNER JOIN ALKALMAZOTT ON RENDSZER.LOGINID=ALKALMAZOTT.AID;**

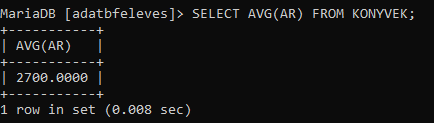
**πnev, jelszo(rendszer ⋈rendszer . loginid = alkalmazott . aidalkalmazott);**

****

1. Készítsünk lekérdezést, amely megadja a könyvek átlagárát! (atlagar)

**SELECT AVG(AR) FROM KONYVEK;**

**πAVG (ar) γAVG (ar)konyvek;**

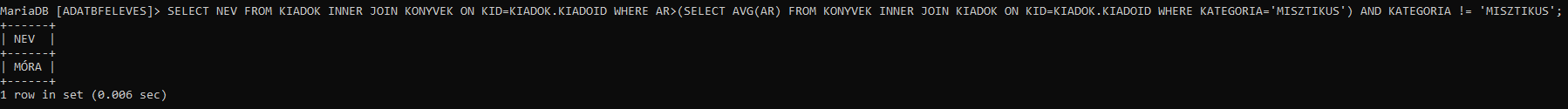


**Bonyolult lekérdezések:**

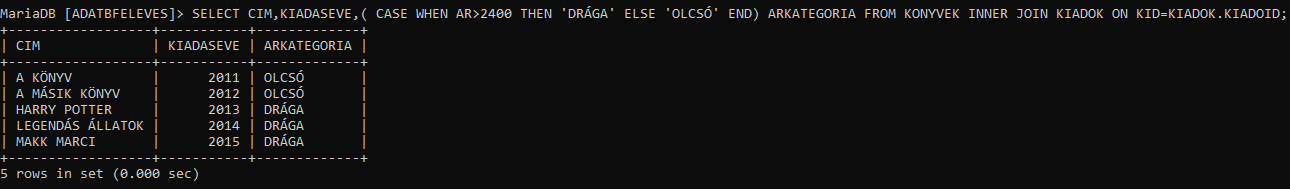
1. Készítsünk lekérdezést, amely megadja bármely „misztikus” műfajú könyv áránál drágább, nem misztikus kategóriájú könyvek kiadójának a neveit! (nev)

**SELECT NEV FROM KIADOK INNER JOIN KONYVEK ON KID=KIADOK.KIADOID WHERE AR>(SELECT AVG(AR) FROM KONYVEK INNER JOIN KIADOK ON KID=KIADOK.KIADOID WHERE KATEGORIA='MISZTIKUS') AND KATEGORIA != 'MISZTIKUS';**

**π nev(kiadok ⋈kid = kiadok.kiadoidalkalmazott) σ ar>( π(AVG(ar)(konyvek⋈kid=kiadok.kiadoid) σKategoria=’Misztikus’);**

1. Készítsünk lekérdezést, amely árkategória szerint csoportosítja a könyvek árait, illetve megjeleníteni azoknak a kiadási évét! (cím,kiadaseve,arkategoria)

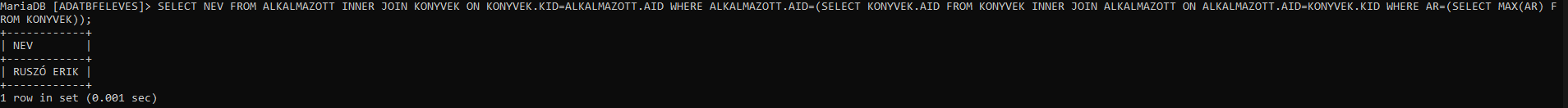
**SELECT CIM, KIADASEVE,( CASE WHEN AR>2400 THEN 'DRÁGA' ELSE 'OLCSÓ' END) ARKATEGORIA FROM KONYVEK INNER JOIN KIADOK ON KID=KIADOK.KIADOID;**



1. Készítsünk lekérdezést, amely megadja annak az alkalmazottnak a nevét, aki a legdrágább könyvet karbantartotta. (nev)

**SELECT NEV FROM ALKALMAZOTT INNER JOIN KONYVEK ON KONYVEK.KID=ALKALMAZOTT.AID WHERE ALKALMAZOTT.AID=(SELECT KONYVEK.AID FROM KONYVEK INNER JOIN ALKALMAZOTT ON ALKALMAZOTT.AID=KONYVEK.KID WHERE AR=(SELECT MAX(AR) FROM KONYVEK));**

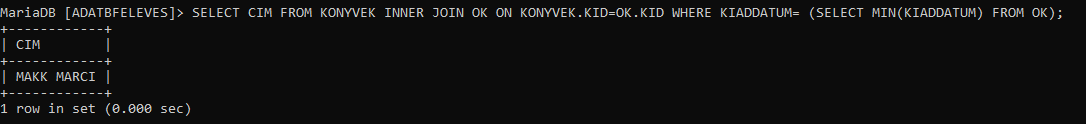
**π nev(Alkalmazott ⋈konyvek.kid = alkalmazott.aidKonyvek) σ alkalmazott.aid=( π konyvek.aid (Konyvek ⋈ alkalmazott.aid=konyvek.kid Alkalmazott) σ ar=( π max(ar)Konyvek);**



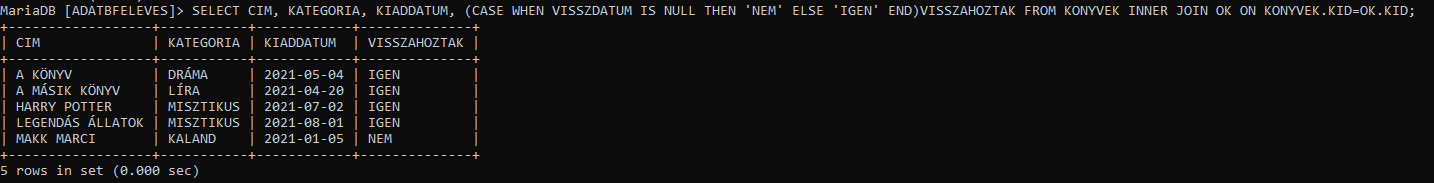
1. Készítsünk lekérdezést, amely megadja annak a könyvek a címét, amely a legrégebben került kikölcsönzésre. (cím)

**SELECT CIM FROM KONYVEK INNER JOIN OK ON KONYVEK.KID=OK.KID WHERE KIADDATUM= (SELECT MIN(KIADDATUM) FROM OK);**

**πcim(konyvek ⋈konyvek . kid = ok . kidok) σ Kiaddatum= (π (MIN(Kiaddatum)OK);**



1. Készítsünk lekérdezést, amely megadja a kikölcsönzött könyvek címét, kategóriáját, kiadásának idejét, és hogy visszahozták-e már őket4 (cím, kategoria, kiaddatum, „visszahoztak”)

**SELECT CIM, KATEGORIA, KIADDATUM, (CASE WHEN VISSZDATUM IS NULL THEN 'NEM' ELSE 'IGEN' END) VISSZAHOZTAK FROM KONYVEK INNER JOIN OK ON KONYVEK.KID=OK.KID;**