Jegyzőkönyv

Adatbázis rendszerek I. Féléves feladat

Készítette: **Bartók-Balogh Gábor**
Nepunkód: **QVTQO8**
Gyak: **Szerda** 14:00-15:30
Vezér: Bednarik László

Féléves feladat 1.rész leírása

Az adatbázis modelljének egy Könyvelő iroda modelljét vettem alapul. Az adatbázis a könyvelő irodával kapcsolatban álló Ügyfelek és szükséges szervezetek adatait fogja tárolni. (ER modell és Relációs Modell) Az adatbázisban a következő egyedek kapnak helyet az alábbi adatokkal:

- **Ügyfelek:**
  a) Adó_azonosító: Elsődleges kulcs (Primary Key) tulajdonságú.
  b) Név
  c) Email_cím: Több értékű tulajdonságú.
  d) lakcím: Összetett tulajdonságú.

  Egy-Több kapcsolat a Könyvelőirodával!

- **Könyvelőiroda:**
  a) Ügyvezető_név
  b) Ir_adószám: Elsődleges kulcs (Primary Key) tulajdonságú.
  c) Székhely: Összetett tulajdonságú.

  Egy-Több kapcsolat az Önkormányzattal, OEP-el és a Nav-val.

- **Önkormányzat:**
  a) Önk_Adószám: Elsődleges kulcs (Primary Key) tulajdonságú.
  b) Ügyintéző: Összetett tulajdonságú.

  Egy-Több kapcsolat a Könyvelőirodával!

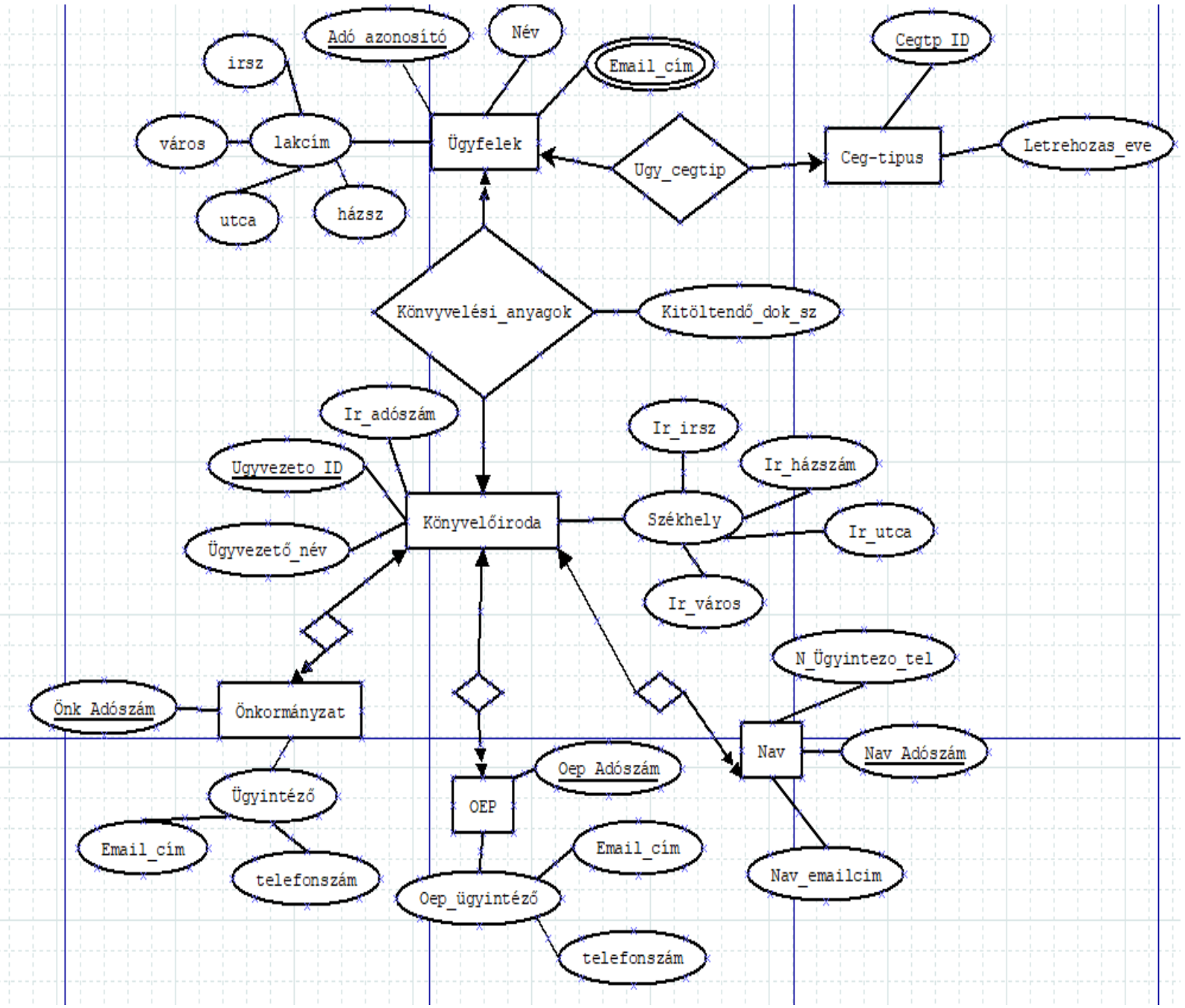
- **OEP**:
  a) Oep_Adószám: Elsődleges kulcs (Primary Key) tulajdonságú.
  b) Oep_ügyintéző: Összetett tulajdonságú.

  Egy-Több kapcsolat a Könyvelőirodával!

- **Nav:**
  a) Nav_emailcím
  b) N_Ügyintéző_tel:
  c) Nav_Adószám: Elsődleges kulcs (Primary Key) tulajdonságú.

  Egy-Több kapcsolat a Könyvelőirodával!

ER modell:


Adó azonosító
Név
irsz
Email_cím
Cegtp ID
város
lakcím
Ügyfelek
Ugy_cegtip
Ceg-tipus
Letrehozas_eve
utca
házsz
Könyvvelési_anyagok
Kitöltendő_dok_sz
Ir_adószám
Ir_irsz
Ugyvezeto ID
Ir_házszám
Könyvelőiroda
Székhely
Ir_utca
Ügyvezető_név
Ir_város
N_Ügyintezo_tel
Önk Adószám
Önkormányzat
Nav
Nav Adószám
Oep Adószám
Ügyintéző
OEP
Email_cím
Email_cím
telefonszám
Oep_ügyintéző
Nav_emailcim
telefonszám

2.rész
Relációs Modell leírása:

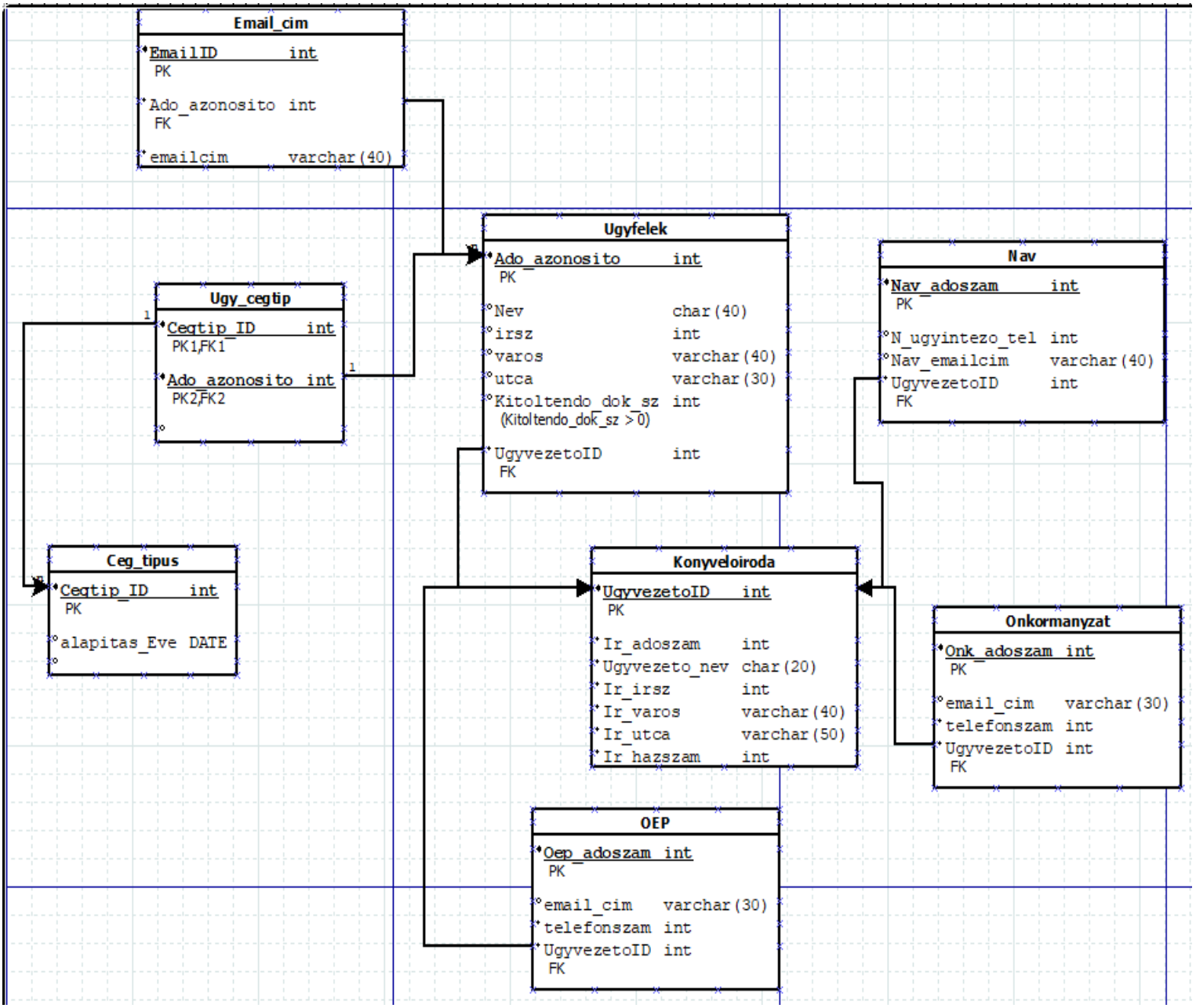
Az ER modell konvertálása Relációs modellre az alábbi módon történik:

Az egyedek az ER modellből külön táblát kapnak melyben tulajdonságaik és azok típusait látjuk azonban a táblák kiegészülnek FK (foreign key) kulcsokkal melyek a kapcsolatot fogját tartani a táblák között. A jelenlegi adatbázis modellben a „Könyvelőirodával" fogja minden tábla tartani a kapcsolatot
így rajta kívül minden más tábla megkapja a Könyvelőiroda PK (primary key) kulcsából származó FK-t. Egy kivételt látunk mégpedig az Ügyfél táblánál, ahol az Email_cím egy külön táblát kapott ennek oka pedig az, hogy az „email_cím" az ER modellben egy többértékű tulajdonság volt melyet Relációs modellnél egy külön táblaként definiálunk ez a tábla ilyenkor megkapja a saját PK kulcsát, illetve az „Ügyfelek" tábla PK kulcsából származtatott FK kulcsot, valamint egy saját tulajdonságot. Az ER modellben látszik, hogy rendelkezünk összetett értékű tulajdonsággal PL: „Ügyfelek" táblánál a „lakcím". Ezeket Relációs táblánál elhagyjuk és külön írjuk le a tulajdonságokat, azaz a „lakcím" helyett irsz, házsz, város, utca tulajdonságok kapnak helyett a táblában.

Az adatbázis relációs sémái:

Ügyfelek: [Ado_azonosito,Nev,irsz,varos, utca,Kitoltendo_dok_sz, UgyvezetoID]
Könyvelőiroda:[UgyvezetoID,Ugyvezeto_nev,Ir_irsz,Ir_varos,Ir_utca,Ir_hazszam]
OEP: [Oep_adoszam,email_cim,telefonszam, UgyvezetoID]
Nav: [Nav_adoszam,N_ugyintezo_tel,Nav_emailcim, UgyvezetoID]
Önkormányzat: [Onk_adoszam,email_cim,telefonszam, UgyvezetoID]
Email_cím: [EmailID,Ado_azonosito,emailcim]

Relációs Modell:



Táblák létrehozása:

Adatbázis létrehozásánál fontos ügyelnünk arra, hogy legelőször mindig azokat a táblákat kreáljuk le, amik nem rendelkeznek FK kulccsal, ezután fogjuk a FK kulccsal rendelkező táblák megalkotását végezni mivel az idegen kulcsok az előzőleg létrehozott táblákra fognak mutatni így csak is akkor lehetséges őket felvinni az adatbázisba ha előtte az elsődleges kulccsal rendelkező táblát
lekreáltuk amire rámutat majd FK kulcsunk. Fontos, hogy FK kulcs generálásánál a kulcs tulajdonsága megegyezzen típusban és méretben is a PK kulcs

tulajdonságaival más esetben hiba miatt nem fogjuk tudni felvinni az adatbázisba a kívánt adatot.

CREATE TABLE Konyveloiroda (Ir_adoszam int PRIMARY KEY, Ugyvezeto_nev char(20) NOT NULL, Ir_irsz int NOT NULL, Ir_varos varchar(40) NOT NULL, Ir_utca varchar(50) NOT NULL, Ir_hazszam int NOT NULL);

CREATE TABLE Ugyfelek(Ado_azonosito int PRIMARY KEY,Nev char(50) NOT NULL,irsz int,varos varchar(40),utca varchar(30),Kitoltendo_dok_sz int,Ir_adoszam int);
ALTER TABLE Ugyfelek **ADD CONSTRAINT FOREIGN KEY**(Ir_adoszam) **REFERENCES** Konyveloiroda(Ir_adoszam);

CREATE TABLE Email_cim(EmailID int PRIMARY KEY,emailcim varchar(40),Ado_azonosito int);
ALTER TABLE Email_cim **ADD CONSTRAINT FOREIGN KEY** (Ado_azonosito) **REFERENCES** Ugyfelek(Ado_azonosito);

CREATE TABLE Nav(Nav_adoszam int PRIMARY KEY,N_ugyintezo_tel int,Nav_emailcim varchar(40),Ir_adoszam int);
ALTER TABLE Nav **ADD CONSTRAINT FOREIGN KEY** (Ir_adoszam) **REFERENCES** Konyveloiroda(Ir_adoszam);

CREATE TABLE OEP(Oep_adoszam int PRIMARY KEY,email_cim varchar(30),telefonszam int,Ir_adoszam int);

ALTER TABLE OEP **ADD CONSTRAINT FOREIGN KEY** (Ir_adoszam) **REFERENCES** Konyveloiroda(Ir_adoszam);

CREATE TABLE onkormanyzat(Onk_adoszam int PRIMARY KEY,email_cim varchar(30),telefonszam int,Ir_adoszam int);
ALTER TABLE Onkormanyzat **ADD CONSTRAINT FOREIGN KEY** (Ir_adoszam) **REFERENCES** Konyveloiroda(Ir_adoszam);

Táblák feltöltése:

INSERT INTO Konyveloiroda **VALUES**(101,123456,'Korpás Haj',3780,'Plankton','Elm utca',20);

INSERT INTO Konyveloiroda **VALUES**(102,123456,'Haj Lakk',3780,'Plankton','Elm utca',20);

INSERT INTO Konyveloiroda **VALUES**(103,123456,'Paprikás Chips',3780,'Plankton','Elm utca',20);

INSERT INTO Ugyfelek **VALUES**(201,'Lila Répa',2450,'Katlan','Perzs utca',3,101);

INSERT INTO Ugyfelek **VALUES**(202,'Zöld Szék',2230,'Pozdorja', 'Fröccs utca',5,101);

INSERT INTO Ugyfelek **VALUES**(203,'Zöld Párna',1212,'Roma','Farkas út',2,103);

INSERT INTO Ugyfelek **VALUES** (204,'SamSamTech',3521, 'Rotty', 'Pézsma út', 6, 102);

INSERT INTO Ugyfelek **VALUES**(205,'Huawei',7643,'Peking','Plazma tér',12,101);

INSERT INTO Ugyfelek **VALUES**(206,'Pézsma Pocok',2313,'Porlo','Ford koz',8,102);

INSERT INTO Ugyfelek **VALUES** (207,'Vajak Istvan',3231, 'Pirkadat','Fiatal ut',1,103);

INSERT INTO Ugyfelek **VALUES** (208,'Dalos Pacsirta', 1775, 'Persely',

'Aphro utca',4,102);

INSERT INTO Ugyfelek **VALUES** (209,'Pikans ovoda es bolcsode', 1453, 'Velezd', 'Keszpent ut',7,101);

INSERT INTO Ugyfelek **VALUES**(210,'COOP abc',5431,'Tur','Semmi koz', 6,102);

INSERT INTO Nav **VALUES**(400,701234567,’valami12@gmail.com’,103);

INSERT INTO OEP **VALUES** (401,'valami32@OEP.hu',06304567893,101);

INSERT INTO Onkormanyzat **VALUES**(402,'valami42@Pest.hu', 06304567678,103);

INSERT INTO Onkormanyzat **VALUES** (403, 'valami43@Borsod.hu', 06304567123,102);

INSERT INTO Onkormanyzat **VALUES** (404, 'valami44@Kiskunhalas.hu', 06304567674,101);

INSERT INTO Email_cim **VALUES**(10,'lilarepa1@gmail.com',201);

INSERT INTO Email_cim **VALUES**(11,'Zoldszek2@gmail.com',202);

INSERT INTO Email_cim **VALUES**(12,'samsamtech3@tech.com',204);

INSERT INTO Email_cim **VALUES** (13, 'pezsma.pocok4@freemail.hu', 206);

INSERT INTO Email_cim **VALUES**(14,'vajak5@freemail.hu',207);

INSERT INTO Email_cim **VALUES**(15,'pikansovibolcsi6@ovi.net',209);

INSERT INTO Email_cim **VALUES**(16,'coopabc7@webmail.hu',210);

Lekérdezések:

1.Email címmel rendelkező ügyfelek lekérdezése.

SELECT u.Nev,e.emailcim **FROM** Ugyfelek u **INNER JOIN** Email_cim e **ON** u.Ado_azonosito = e.Ado_azonosito;

$\pi_{u.nev,\ e.emailcim}(\rho_{u}\ ugyfelek \bowtie_{u.ado\_azonosito = e.ado\_azonosito} \rho_{e}\ email\_cim)$

2. Hatnál több kitöltendő dokumentummal rendelkező ügyfelek neve szám szerint növekvő sorrendben:

SELECT Nev,Kitoltendo_dok_sz **FROM** Ugyfelek **WHERE** Kitoltendo_dok_sz > 6 **ORDER BY** Kitoltendo_dok_sz;

$\tau_{kitoltendo\_dok\_sz} \pi_{nev,\ kitoltendo\_dok\_sz} \sigma_{kitoltendo\_dok\_sz > 6}\ ugyfelek$

3. Ügyfelek számának lekérdezése:

SELECT COUNT(Nev) **AS** ugyfelekszama **FROM** Ugyfelek;

$\pi_{COUNT\ (nev) \rightarrow\ ugyfelekszama} \gamma_{COUNT\ (nev)}\ ugyfelek$

4. p betűvel kezdődő városok neve:

SELECT varos **FROM** Ugyfelek **WHERE** varos **LIKE** 'p%';

$\pi_{varos} \sigma_{varos\ LIKE\ "p\%"}\ ugyfelek$

5.Kitöltendő összes dokumentum száma

SELECT sum(Kitoltendo_dok_sz) **FROM** Ugyfelek;

$\pi_{SUM\ (kitoltendo\_dok\_sz)} \gamma_{SUM\ (kitoltendo\_dok\_sz)}\ ugyfelek$

1.Email címmel nem rendelkező ügyfelek lekérdezése

SELECT u.Nev **FROM** Ugyfelek u **WHERE** u.Ado_azonosito **NOT IN**(SELECT Ado_azonosito FROM Email_cim) **GROUP BY** Nev;

$\pi_{u.nev} \rho_{u}\ ugyfelek\ \sigma_{u.Ado\_azonosito != \pi_{ado\_azonosito}\ email\_cim}$

2.Ügyvezetők kódjának és nevének illetve az ügyfeleiknek a

számának lekérdezése:

SELECT k.UgyvezetoID,k.Ugyvezeto_nev, **COUNT(DISTINCT**(Ado_azonosito)**) FROM** Konyveloiroda k,Ugyfelek u **WHERE** k.UgyvezetoID = u.UgyvezetoID **GROUP by** UgyvezetoID;

$\pi_{k.\,ugyvezetoid,\,k.\,ugyvezeto\_nev,\,COUNT\,(\backslash delta\,ado\_azonosito)}\gamma_{ugyvezetoid,\,COUNT\,(\backslash delta\,ado\_azonosito)}\sigma_{k.\,ugyvezetoid\,=\,u.\,ugyvezetoid}(\rho_{k}\,konyveloiroda \times \rho_{u}\,ugyfelek)$

3.Ügyvezetők ügyfeleinek száma és kitöltendő dokumentumainak száma Ügyvezetőkre bontva:

SELECT k.UgyvezetoID,count(DISTINCT u.Ado_azonosito) **AS** Ugyfelszam**,sum**(u.Kitoltendo_dok_sz) **FROM** Konyveloiroda k **LEFT OUTER JOIN** Ugyfelek u **ON** k.UgyvezetoID = u.UgyvezetoID **GROUP BY** k.UgyvezetoID;

$\pi_{k.\,ugyvezetoid,\,COUNT\,(\backslash delta\,ado\_azonosito)\rightarrow\,ugyfelszam,\,SUM\,(kitoltendo\_dok\_sz)}\gamma_{ugyvezetoid,\,SUM\,(kitoltendo\_dok\_sz),\,COUNT\,(\backslash delta\,ado\_azonosito)}(\rho_{k}\,konyveloiroda \bowtie^{oL}_{k.\,ugyvezetoid\,=\,u.\,ugyvezetoid} \rho_{u}\,ugyfelek)$

4.Oep ügyfelek lekérdezése:

SELECT u.Nev **FROM** Ugyfelek u **WHERE** UgyvezetoID=**(SELECT** UgyvezetoID **FROM** Oep**);**

$\pi_{u.\,nev}\sigma_{ugyvezetoid\,=\,\pi_{ugyvezetoid}\,oep}\,\rho_{u}\,ugyfelek$

5.Ügyfelek adatai és a szervezetek elérései ahova tartoznak.

SELECT u.UgyvezetoID,u.Nev, o.telefonszam, n.N_ugyintezo_tel,onk.telefonszam,onk.email_cim **FROM** Ugyfelek u **LEFT JOIN** Oep o **ON** u.UgyvezetoID = o.UgyvezetoID **LEFT JOIN** Nav n **ON** u.UgyvezetoID = n.UgyvezetoID **LEFT JOIN** Onkormanyzat onk **ON** u.UgyvezetoID **=** onk.UgyvezetoID **WHERE** u.UgyvezetoID **IN** (101,102,103);