# 9. gyakorlat (4. 7.)

Jelszó a Neptun kódom, kisbetűvel

## Berezvai Dániel jegyzete <http://elte.3ice.hu/>

## Kapcsolatok leképezése

08ab1e02.pdf 32. oldal

1. változat: Ha egy olvasónak egyszerre csak egy könyvet

adnak ki, akkor a kölcsönzés 1:1 kapcsolatot jelent. Ilyenkor a

KÖLCSÖN sémában a könyvszám és az olvasószám egyaránt

kulcs. Továbbá, a visszahozás attribútumra nincs szükségünk,

mivel a könyv visszahozásával a könyv-olvasó kapcsolat

megszűnik.

Tehát, a

KÖLCSÖN (könyvszám, olvasószám, kivétel)

vagy a

KÖLCSÖN (könyvszám, olvasószám, kivétel)

sémát vehetjük fel a kapcsolathoz.

A KÖLCSÖN sémát az azonos kulcsú sémába olvasztva a

KÖNYV (könyvszám, szerző, cím, olvasószám, kivétel)

OLVASÓ (olvasószám, név, lakcím)

vagy a

KÖNYV (könyvszám, szerző, cím)

OLVASÓ (olvasószám, név, lakcím, könyvszám, kivétel)

adatbázissémákat kapjuk.

2. változat: Ha egy olvasó több könyvet

is kikölcsönözhet, akkor a könyv-olvasó

kapcsolat N:1 típusú. Ekkor a KÖLCSÖN

sémában csak a könyvszám lehet kulcs,

ezért a KÖLCSÖN sémát csak a KÖNYV

sémába olvaszthatjuk:

KÖNYV (könyvszám, szerző, cím, olvasószám, kivétel)

OLVASÓ (olvasószám, név, lakcím)

3. változat: Ha az egyes könyvek korábbi

kölcsönzéseit is nyilvántartjuk, akkor nem csak egy

olvasóhoz tartozhat több könyv, hanem egy

könyvhöz is több olvasó (N:M kapcsolat), sőt adott

olvasó adott könyvet egymás után többször is

kikölcsönözhet. Ezért a KÖLCSÖN sémában

{könyvszám, kivétel}

vagy

{könyvszám, visszahozás}

a kulcs, a KÖLCSÖN táblát most sem a KÖNYV, sem

az OLVASÓ táblába nem tudjuk beolvasztani. Az

adatbázisséma ezért a következő:

KÖNYV (könyvszám, szerző, cím)

OLVASÓ (olvasószám, név, lakcím)

KÖLCSÖN (könyvszám, olvasószám, kivétel, visszahozás)

# SQL DDL, DML áttekintése, nézetek, inline nézetek

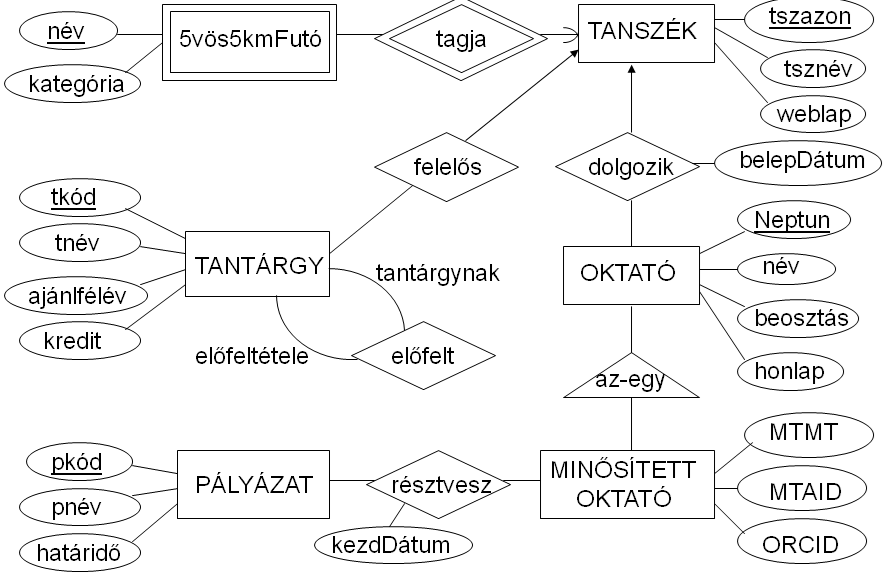
http://people.inf.elte.hu/sila/edu14feb/ab1\_gy07.html

## E/K diagram és leképezése relációs modellre - Egy könyvtár adatmodellje

Téglalap – egyedek

Rombuszok – kapcsolatok

Ellipszisek – attribútumok



Háromszögben "az-egy" = "IS A" kapcsolat.

Minősített oktató esetén nincs is ezért elsődleges kulcs.

### SQL séma

…

## DDL: CREATE, DROP, ALTER TABLE/VIEW

Adat definíciós utasítások (Data definition language)

Ebből kettő volt múlt órán, egy kimaradt:

### ALTER TABLE

Oszlopok (attribútumok) hozzáadása, módosítása, és törlése.

Megszorítások szerkesztése, átnevezése, kidobása, kikapcsolása.

## Tranzakció kezelés

Következő félévben…

Érvényesítés: COMMIT (Amit eddig csináltunk, véglegesen a háttértárolón tárolódjon.)

Visszagörgetés: ROLLBACK (Alapból csak az utolsó COMMIT-ig lehet visszavonni.)

Mentési pontok: SAVEPOINT (Névvel, utána ROLLBACK TO savepoint működik.)

## SQL DML (INSERT, DELETE, UPDATE)

Data manipulation language

* Adatok karbantartása adattáblán illetve nézeten keresztül, és megfigyelni ezek egymásra hatását: Hogyan hat a nézetek adatainak módosítása az adattáblára, és a másik irányban, az adatok módosítása hogyan jelenik meg a nézetben?
* Eközben mentési pontokat adjunk meg, ahova folyton visszagörgetjük (visszaállítjuk a táblák eredeti állapotát).

Kétféle INSERT van.

INSERT INTO tábla [(Mezőnevek)] VALUES (Értékek)

Nem kötelező megadni a mezőneveket, ha megadjuk sem kötelező az összes mezőt megadni.

Egy sor:

INSERT INTO sz (n, gy) VALUES ('róka', 'sajt');

INSERT INTO sz (gy, n) VALUES ('sajt', 'róka');

INSERT INTO sz (n) VALUES ('róka');

Több sor:

INSERT INTO sz (n) SELECT ename FROM emp;

15 név egyből bekerül. NULL gyümölcsöt szeretik.

INSERT INTO sz SELECT ename, job FROM emp;

15 sor megint, bár gyümölcs (gy) helyett foglalkozás (job).

UPDATE sz SET gy = 'üzletkötő' WHERE gy='SALESMAN';

A komissziót adjuk hozzá a fizetésekhez, majd nullázzuk:

UPDATE emp SET sal = sal + NVL(comm, 0), comm=NULL;

Visszavonás:

ROLLBACK;

Törlés:

DELETE FROM sz WHERE gy='alma';

DELETE FROM sz WHERE gy IS NULL;

## Oracle specialitások

Felső-N elemzés ROWNUM segítségével:

SELECT \* FROM sz WHERE ROWNUM <= 5;

Ez az első öt. (Máshol LIMIT.)

Csillag mellé nem fér ROWNUM, megoldás: táblanév.\*

SELECT ROWNUM, sz.\* FROM sz;

Ötödiket kiírni nehéz, mert a ROWNUM = 5 nem működik.

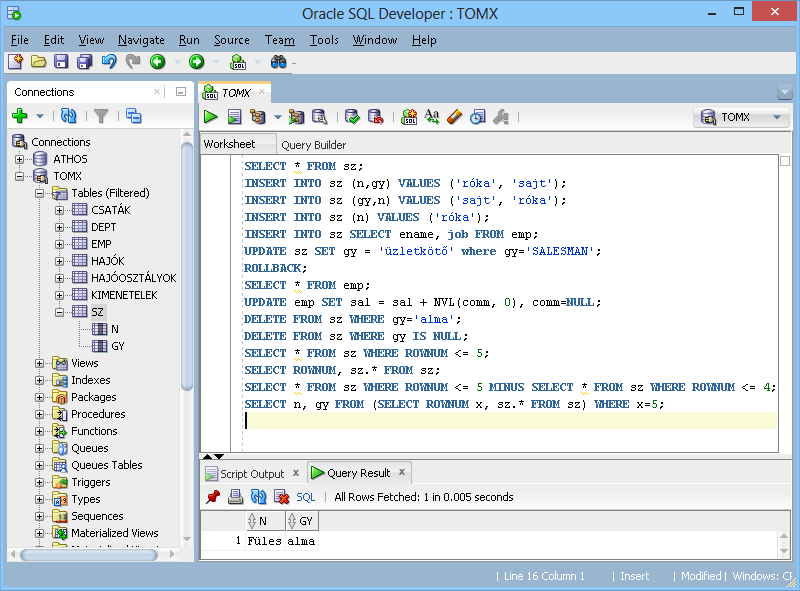
Kivonással: SELECT \* FROM sz WHERE ROWNUM <= 5 MINUS SELECT \* FROM sz WHERE ROWNUM <= 4;

SQL-lel:

SELECT \* FROM (SELECT ROWNUM x, sz.\* FROM sz) WHERE x=5;

Átnevezzük a ROWNUM-ot x-re. Ha nem kell a sorszám:

SELECT n, gy FROM (SELECT ROWNUM x, sz.\* FROM sz) WHERE x=5;



SELECT \* FROM sz;

INSERT INTO sz (n,gy) VALUES ('róka', 'sajt');

INSERT INTO sz (gy,n) VALUES ('sajt', 'róka');

INSERT INTO sz (n) VALUES ('róka');

INSERT INTO sz SELECT ename, job FROM emp;

UPDATE sz SET gy = 'üzletkötő' where gy='SALESMAN';

ROLLBACK;

SELECT \* FROM emp;

UPDATE emp SET sal = sal + NVL(comm, 0), comm=NULL;

DELETE FROM sz WHERE gy='alma';

DELETE FROM sz WHERE gy IS NULL;

SELECT \* FROM sz WHERE ROWNUM <= 5;

SELECT ROWNUM, sz.\* FROM sz;

SELECT \* FROM sz WHERE ROWNUM <= 5 MINUS SELECT \* FROM sz WHERE ROWNUM <= 4;

SELECT n, gy FROM (SELECT ROWNUM x, sz.\* FROM sz) WHERE x=5;

# PL/SQL

Két programot egymás után nem tudunk lefuttatni. Erre megoldás: Program → jelezzük, hogy vége van → program. Vagy Programba rakjuk a programokat

[≪Verzió≫]

DELARE

BEGIN

[EXCEPTION] (Hibakezelés)

END;

## Blokk

Bárhol.

## Mostantól van logikai (BOOLEAN) típus

Eddig (algebra, SQL) nem volt.

## Változók

blokk előtt: felhasználói, környezeti

blokkon belül: felhasználói, környezeti, PL/SQL

blokk után: felhasználói, környezeti

Írásmód: vagy .

Példa: SELECT \* FROM &x; → Program megkérdezi, melyik táblából SELECT-álja a csillagot.

Tipikus használat: jelszó bekérés.

Példa 2: SELECT \* FROM &&x; → Program először megkérdezi, a jövőben soha többet. &x és &&x esetén is emlékszik.

Kiírás: SELECT '&x' FROM DUAL;

Milyen változóink vannak: DEFINE vagy DEF

Törlés: UNDEFINE x vagy UNDEF x.

Bekérés: ACCEPT vagy ACC

Bekérés kiírással: ACCEPT x prompt 'Kérem az x-et'

Az ACCEPT letörli, és újra létrehozza a változót.

ACCEPT x prompt 'Kérem az x-et' hide

Beszédes változó név helyette:

SELECT \* FROM &kerek\_egy\_adatbazis\_nevet;

var x number

print x

:x≔

felhasználói az &változónév

környezeti a :változónév

harmadik a változónév

## Utasítások

NULL – üres utasítás, nem csinál semmit

GOTO – ugrás címkéhez. Végéhez ugrani nem tudunk, de az END elé rakott NULL utasításnak már adhatunk címkét, az működik.

Értékadás kétféle:

≔  
vagy  
SELECT kifejezéslista INTO változólista (vagy rekordváltozó) FROM …

Elágazás: ELSIF van ELSEIF helyett.

CASE [WHEN THEN] [ELSE] END CASE

Ciklus:

Végtelen, elől és hátul- tesztelős, számlálós vagy halmazszerű

LOOP (…) END LOOP;

WHILE feltétel LOOP utasítások END LOOP;

FOR ciklusváltozó IN [REVERSE] alsóhatár .. felsőhatár LOOP utasítások END LOOP;

EXIT [címke] [WHEN feltétel] - ciklusból kilép.

Gyakorlat vége.

Változók következő gyakorlaton.