



SZÉCHENYI TERV

RADVÁNYI TIBOR

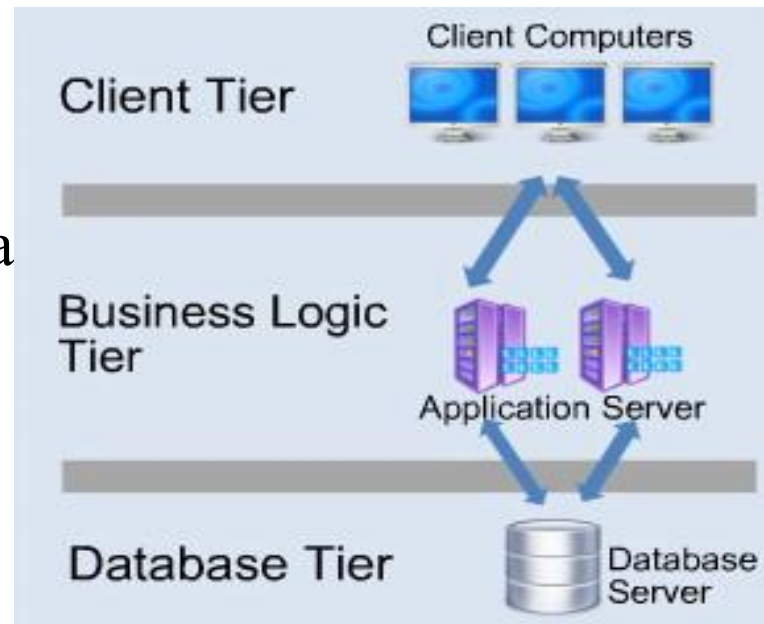
EKF TTK Számítástudományi Tanszék

Adatbázisrendszerek

**Alkalmazható természettudományok oktatása a tudásalapú
társadalomban – TÁMOP-4.1.2.A/1-11/1-2011-0038**

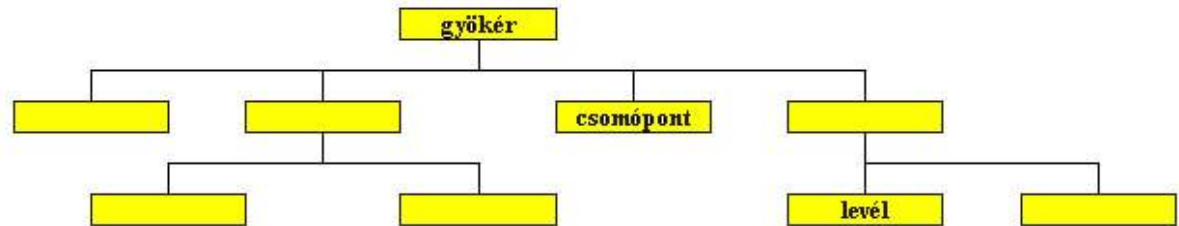
Alapfogalmak, adatbázis-kezelő rendszerek

- Adat és információ
- Az adatbázis
- Lokális
- Fájl-szerver architektúra
- Kliens-szerver architektúra
- Multi Tier architektúra

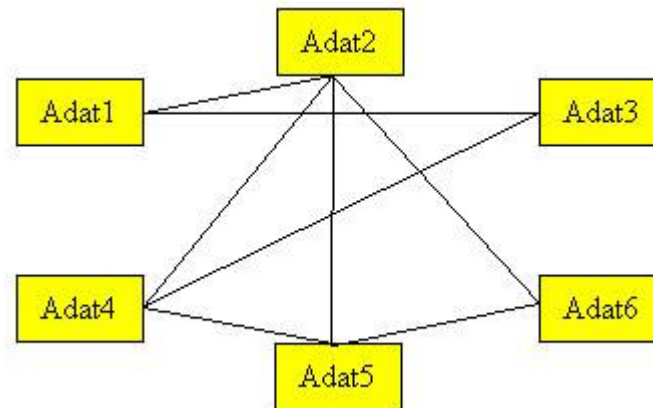


Adatmodellek és fejlődésük

- Hierarchikus modell



- Hálós modell



- Relációs modell

	Oszlop			
Sor				
			Mező	

Adatbázisok tervezése, eszközök

- Normalizálás
- Normálformák
- Számítógéppel támogatott tervezés
- A MySQL WorkBench



Eszközök



Data Modeling

Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.



Open Existing EER Model

Or select a model to open or click here to browse.

- feladat02**
Last modified Wed Oct 17 10:14:16 2012
- feladat01**
Last modified Wed Oct 17 09:58:30 2012
- focicsapat**
Last modified Wed Oct 10 10:23:25 2012
- tanulok**
Last modified Wed Oct 10 09:48:04 2012
- kolcsonzes**
Last modified Wed Oct 03 10:25:29 2012
- dolgozo**
Last modified Wed Oct 03 07:08:42 2012
- db**
Last modified Wed Oct 03 06:44:38 2012



Create New EER Model

Create a new EER Model from scratch.



Create EER Model From Existing Database

Create by connecting and reverse engineering.



Create EER Model From SQL Script

Import an existing SQL file.

Itt található a már elkészített modellek, melyeket újra megnyithatunk és változtathatjuk.

Erre kattintva új modellt hozhatunk létre

Az SQL nyelv alapjai

- A Data Definition Language
- A Data Manipulation Language
- A Data Control Language
- A Query Language

```
CREATE TABLE testtab
  (col1 VARCHAR2(10))
  STORAGE (INITIAL 100K
           NEXT 50K
           MINEXTENTS 1
           MAXEXTENTS 99
           PCINCREASE 20);
```

```
UPDATE emp_copy2 SET e_name='Kovács' WHERE emp_no=7499;
```

```
UPDATE emp_copy2 ec SET (emp_no, e_name) =
  (SELECT emp_no, e_name FROM emp e WHERE e.emp_no=ec.emp_no)
WHERE e_name LIKE 'I%';
```

Az SQL nyelv alapjai – A QL

Dolgozo				
ID	Neve	Szdatum	Varos	Fizetes
1				
2	Kiss Szilárd	1985.01.05	Eger	120000
1				
3	Nagy József	1973.05.06	Eger	156000
1				
5	Gipsz Jakab	1955.11.01	Miskolc	210000
1				
7	Kovács Piroska	1996.07.15	Budapest	189000



```
SELECT Neve, Fizetes FROM Dolgozo  
WHERE Fizetes >= 150000
```

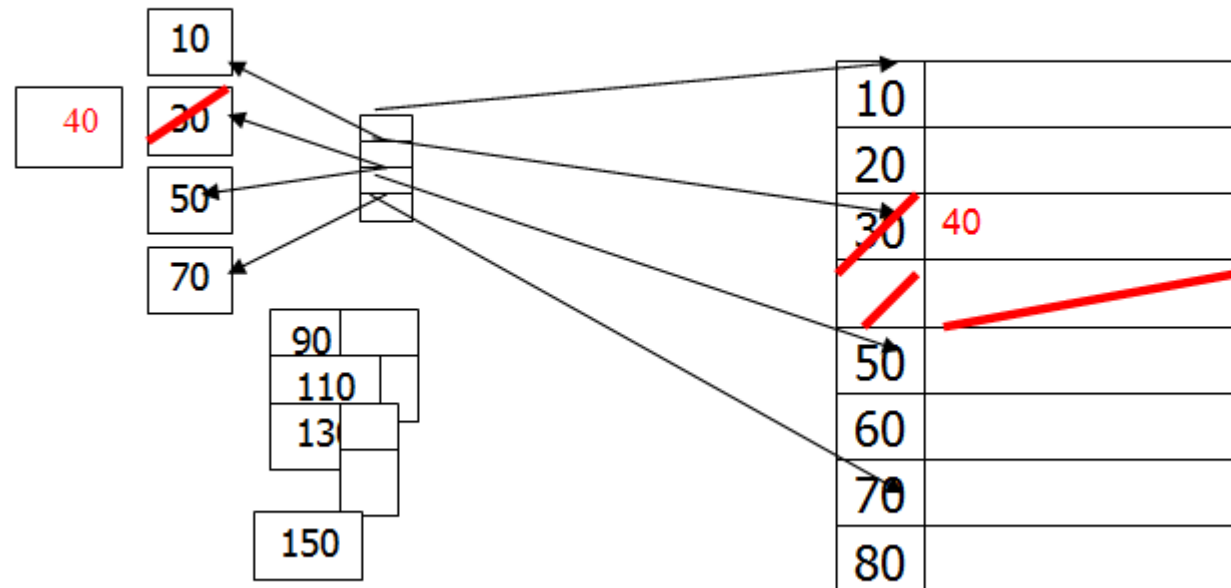


Neve	Fizetes
Nagy József	156000
Gipsz Jakab	210000
Kovács Piroska	189000

Nézetek és indexek

- Megszorítások és integritási feltételek
- Triggerek
- Kulcsok és használatuk

Töröljük a 30-as rekordot!



Triggerek

```
CREATE TRIGGER dolg_szamlal
AFTER INSERT ON dolgozo
FOR EACH ROW
WHEN (NEW.fizetes < 100000)
BEGIN
UPDATE szamlalo SET ertek=ertek+1;
END;
```

```
CREATE TRIGGER dolg_kezdo
BEFORE INSERT ON dolgozo
BEGIN
UPDATE szamlalo SET ertek=0;
END;
```

A PL/SQL alapjai

- Egyszerű típusok
- Összetett típusok
- Vezérlési szerkezetek

```
DECLARE
v_Faktorialis NUMBER(5);
i PLS_INTEGER;
BEGIN
i := 1;
v_Faktorialis := 1;
LOOP
v_Faktorialis := v_Faktorialis * i;
i := i + 1;
END LOOP;
EXCEPTION
WHEN VALUE_ERROR THEN
DBMS_OUTPUT.PUT_LINE(v_Faktorialis
|| ' a legnagyobb, legfeljebb 5-jegyű faktoriális.');
```

A PL/SQL változói

```
név típus [NOT NULL] {:=|DEFAULT} kifejezés;
```

```
DECLARE
  -- NOT NULL deklarációnál kötelező az értékadás
  v_Szam1 NUMBER NOT NULL := 10;

  -- v_Szam2 kezdőértéke NULL lesz
  v_Szam2 NUMBER;

BEGIN
  v_Szam1 := v_Szam2; -- VALUE_ERROR kivételt eredményez
END;
```

Köszönöm megtisztelő figyelmeteket!