# 1. előadás

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

Hétfőn, kedden kezdés: 8:30

Vége: 10:00

Hétfőn előadó: Benczúr András

Kedden előadó: Hajas Csilla [sila@inf.elte.hu](mailto:sila@inf.elte.hu)

Előadás kedden is van. Mindenki oda jár, ahova szeretne. Én mindkettőre fogok, sokan várhatóan egyikre se.

Gyakorlatról infó: Jövőre – a bukóknak – kötelező lesz az előadáson a részvétel, lesz jelenléti ív. Idén nem.

Hétfőn késtem, kedden a jegyzetből hiányzó dolgokat pótoltam.

A két előadás egyforma lesz.

(Update: Nem egyforma eddig…)

# Hétfői előadás: Relációs adatbázis-kezelés alapjai

## Adatbázis

Számítógéppel kezelt adatok gyűjteménye

## Adatmodell

* Számítógép számára hogyan néz ki az adat
* Felhasználó számára hogyan néz ki az adat

# Relációs adatmodellel

## Gyűjtemény

Reláció: Sor típusú adatokat gyűjt

Sor típus egy n-es:

Reláció sémája (mert nevet kell adnunk): Relációnév (sortípus)

Reláció előfordulása: A sortípusának megfelelő véges sok sorból áll.

Relációs adatbázis sémája: relációsémák

Ha röviden akarunk hivatkozni:

Előfordulása - példánya: az egyes relációk előfordulásai

Kaptunk papírt, rajta sémákkal.

## Szemléltetés

Tábla vagy táblázat formában képzeljük el.

Angol terminológiában szinonimák a Relation és a Tábla.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Másféle szemléltetés: Sor, mint függvény: Egy sor az attribútum nevekhez rendel értékeket.

## Korlátozom

Ha korlátozom a függvényt halmazra, ami részhalmaza halmaznak, halmaz az indexű attribútumokból, akkor a sornak az halmazra való megszorítását úgy kapjuk meg, hogy -hez fog tartozni a sornak a…

Sorból könnyen tudjuk képezni egy bizonyos részét. Veszem a megfelelő oszlopban előforduló értékeket.

Most nézzük meg, tudjuk-e értelmezni a kiosztott példát!

## Kulcs

Aláhúzott attribútum név mit jelent: Példában egy speciális attribútum-párt jelent. Egyértelműen meghatározza a filmet a címe és évszáma. Nincs két különböző sor, ahol a cím és év megegyezik egymással. (Lehet két Mátrix című film, de nem ugyan abban az évben.)

Mindjárt a séma megadásánál meg lehet adni.

(részhalmaza)

kulcs: megszorítás. Olyan előfordulásokat kezelünk / fogadunk el, amelyben ha van két sor, és , (használjuk az előfordulás jelölést: )  
és megszorítva -re, (azaz ) ugyan az, mint , akkor abból következik, hogy (Vagyis nincs két olyan sor, aminek minden kulcsa páronként megegezik.)

Egyben:

## Idegenkulcs hivatkozás

Pl.: A hivatkozik a re is, másrészt hivatkozik a re is.

Formálisan: Két reláció.

Egyik: reláció, kulcs

reláció, hivatkozásokat tartalmazó lista

idegen kulcs és hivatkozik reláció kulcsára.

sor eleme relációnak:

akkor létezik sor, úgy, hogy

Lényeges megadni és rendezettségét is.

## Mit kezdünk az adatokkal?

Erre relációs adatbázis-kezelő nyelveket fogunk megismerni:

* Metaadatok kezelése – leíró, segéd
* Adatmanipulációs nyelvek – lekérdezések, adatmódosítások

# Lekérdezések

Lekérdező nyelvek magadása két részből áll:

Szintaktikus alak, ami használja az adatbázis sémát

Szemantika – kiértékelési eljárásokat ad meg

Kiosztott lapon most nézzük a részt. Alatta kérdések:

Próbáljunk olyan lekérdező nyelvet adni, ami segítségével, számítógéppel egyértelműen ki tudjuk értékelni az összes kérdést.

Három ilyen nyelvet fogunk megismerni:

* Relációs algebra
* SQL lekérdező nyelv
* (Datalog - Logikai alapnyelv)
* ((todo))

# Relációs algebra

Műveleteket ad meg. Tábla előfordulásokból táblát állít elő

Táblából táblát állít elő, amit újra használhatunk, összetett műveletekben.

## Műveletek gyakorisági sorrendben:

Relációnevek és .

Sémák: és

## 1 Kiválasztás – szelekció

Jele a kis szigma, nekem helyette jó a sima betű.

a feltétel.

Elemi feltételek:

egy művelet, lehet: egyenlő, nem egyenlő, , és . todo.

Összetett feltételek: és, vagy, negáció, zárójelezés.

## 2 Vetítés - projekció

Oszlopok szűrése

Jele a pi, nekem is jó a \pi: .

Az eredmény halmaz. Azonos sorok is megjelenhetnek.

## Szorzás jellegű műveletek

Több reláció összefűzése.

### 3 Direkt szorzat

Jele a szorzás -e.

( a konkatenáció jele)

Attribútumok egyértelmű megadásához jelölés, így todo is egyértelművé válik.

### 4 Natural Join (Természetes összekapcsolás)

Jele a függőleges vonalak közötti X (szorzás -e), szerintem "csokornyakkendő" Nekem jó a \bowtie: . (Vagy ⋈.)

Adott

## 5 Halmazműveletek

(ez azt jelenti, hogy ugyan az a sémájuk)

### Unió

### Különbség

Mindegyik művelet monoton nő, kivéve a Különbség. (Ha a bemenetekhez sort adunk, nő az eredmény sorainak száma. Csak akkor nem, ha a kivont táblázathoz adunk sort.

### (Átnevezés)

Ez akkor kell, ha a -t választjuk helyett. (Natural Join, direkt szorzat helyett.)

Jele a ró, nekem is jó a \rho:

# Kifejezések

Adatbázis séma relációnevei. Ezek a változók.

Jövő héten.

Előadás vége.

# Keddi előadás

A honlapon, <http://people.inf.elte.hu/sila/edu14feb/AB1EA.html> jelszavas anyagok vannak.

Felhasználónév: hallgato, jelszó: eltegyak

Egyelőre a honlap <http://people.inf.elte.hu/sila/edu13feb/AB1EA.html>

## Mivel foglalkozunk, mi a célunk

* Adatbázisok-1: Relációs adatbázis-kezelés megismerése, relációs adatmodell, algebrai és logikai lekérdező nyelvek, SQL, PL/SQL, E/K modell, relációs adatbázisok tervezése.
* Erre épül BSc-n Adatbázisok-2, valamint több MSc kurzus
* Tudnivalók az előadásról, tematika, előadás bemutatók, és tudnivalók a vizsgáról és a vizsgatematika: <http://people.inf.elte.hu/sila/edu14feb/AB1EA.html>
* A gyakorlat az előadáson elhangzottakra támaszkodik, gyakorlat tematikája, feladatsorok, segédanyagok: <http://people.inf.elte.hu/sila/edu14feb/AB1GY.html>
* A gyakorlat felépítése miatt az adatbázis-kezelő rendszerek felépítése később lesz, a relációs lekérdezésekkel kezdünk.

## Könyv

Ullman-Widom: Adatbázisrendszerek Alapvetés - Második átdolgozott kiadás, Panem, 2009

## További hasznos linkek

* Oracle Junior képzés, szemináriumsorozat hallgatóknak: <http://www.houg.hu/juniorkepzes>
* Magyarországi Oracle Felhasználók Egyesülete: <http://www.houg.hu/>

## Ki ismeri az SQL-t?

Én!

* Ki látott már relációs adatbázist vagy relációt? 1. példa: A jelentkezési adatok egyeztetésére táblázat (lásd papíron a jelenléti ív /csak az első előadáson van)
* Keresés természetes azonosító (név) szerint vagy mesterséges azonosító (neptunkód) szerint melyik jobb?
* Hogyan könnyíthető meg a keresés? Jelentkezési dátum szerint rendezve, vagy ha névsorban vannak az adatsorok?
* Ki ismeri az SQL-t? Itt mi a különbség?

(1) SELECT B FROM R WHERE A<10 OR A>=10;

(2) SELECT B FROM R;

|  |  |
| --- | --- |
| R | |
| A | B |
| 5 | 20 |
| 10 | 30 |
| 20 | 40 |
| … | … |

A különbség az, hogy (1) csak azokat a sorokat adja vissza, amelyekben az A oszlopban szám van. (2) a NULL-t tartalmazó sorokat is kiválasztja.

(később lesz háromértékű logika: T/F/U)

## Mi is az adatmodell? (később lesz bővebben E/K)

* Az adatmodell a valóság fogalmainak, kapcsolatainak, tevékenységeinek magasabb szintű ábrázolása
* Kettős feladat: az adatmodell megadja, hogy a számítógép számára és a felhasználó számára hogy néznek ki adatok.

Az adatmodell adatok leírására szolgáló jelölés.

Ez a leírás általában az alábbi három részből áll:

1. Az adat struktúrája (később: fizikai és fogalmi adatmodell)
2. Az adaton végezhető műveletek (lekérdezések, módosítások, feldolgozások legyenek megfogalmazhatók)
3. Az adatokra tett megszorítások (milyen adatokat engedélyezünk? később: megszorítások, triggerek)

Implementáció kérdései: Nagy tömegben hatékony és sok felhasználó számára biztonságos legyen a megvalósítás.

## A fontosabb adatmodellek

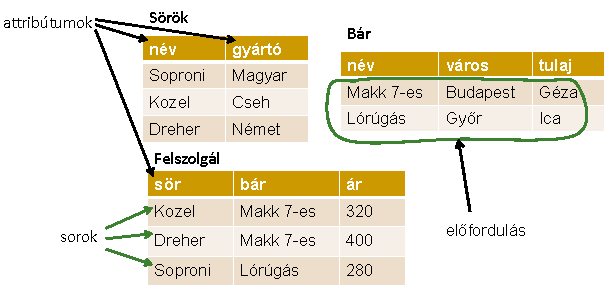
* Hálós, hierarchikus adatmodell (apa-fiú kapcsolatok gráfja, hatékony keresés)
* Relációs adatmodell (táblák rendszere, könnyen megfogalmazható műveletek)
* Objektum-orientált adatmodell (az adatbázis-kezelés funkcionalitásainak biztosítása érdekében gyakran relációs adatmodellre épül)
* Logikai adatmodell (szakértői rendszerek, tények és következtetési szabályok rendszere)
* Dokumentum típusú adatok, félig strukturált adatmodell (XML)

## Példa relációkra (első példa: lásd jelenléti ív)

Itt az adatokat természetes, táblázatos formában kezeljük.

Egy reláció sémája:

Az adatbázis sémája:



## Relációs adatmodell története

Akit érdekel…

* E.F. Codd 1970-ban publikált egy cikket A Relational Model of Data for Large Shared Data Banks Link: http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf amelyben azt javasolta, hogy az adatokat táblázatokban, relációkban tárolják. Az elméletére alapozva jött létre a relációs adatmodell, és erre épülve jöttek létre a relációs adatmodellen alapuló relációs adatbázis-kezelők.
* Relációs (objektum-relációs) adatbázis-kezelők például: ORACLE , INFORMIX , SYSBASE , INGRES, DB2, stb
* Relációs lekérdező nyelvek, 3 nyelv lesz (gyakorlaton is)
  + Relációs algebra (algebrai megközelítés, kiértékelési eljárások),
  + Datalog (logika alapú megközelítés, összetett kérdésekben segít),
  + SQL szabvány relációs lekérdező nyelv (SQL és SQL/PSM lesz).

## Relációs adatmodell előnyei

Miért ez a legelterjedtebb és legkifinomultabb?

* Az adatmodell egy egyszerű és könnyen megérthető strukturális részt tartalmaz. A természetes táblázatos formát nem kell magyarázni, és jobban alkalmazható.
* A relációs modellben a koncepcionális – logikai – fizikai szint teljesen szétválik, nagyfokú logikai és fizikai adatfüggetlenség. A felhasználó magas szinten, hozzá közel álló fogalmakkal dolgozik (implementáció rejtve).
* Elméleti megalapozottság, több absztrakt kezelő nyelv létezik, például relációs algebra (ezen alapul az SQL automatikus és hatékony lekérdezés optimalizálása).
* Műveleti része egyszerű kezelői felület, szabvány SQL.

## Relációs adatmodell – relációs séma

* Adatok gyűjteményét kezeli (gyűjtemény azonosítása: név) A gyűjtemény - reláció (tábla, táblázat) megadása
* A gyűjtemény milyen típusú adatokat gyűjt? adattípus: sor-típus. A sor-típus megadása:  
  röviden
* Relációséma: Relációnév (sortípus) (itt: kerek zárójelben) vagyis  
  röviden jelöléssel
* PÉLDA: lásd jelenléti ív fejléce

## Relációs adatmodell – előfordulás (1/2)

todo

Relációs adatmodell – elıfordulás (1/2)

• Mit jelent egy konkrét sor?

sor <A1: érték1, …, Ak: értékk>

• Reláció előfordulás (példány, instance)

A sor-típusnak megfelelő véges sok sor (sorok halmaza).

Példa: Jelenléti ív táblázat (160 sor)

{t1, ... ,tm} ahol ti (tuple, sor, rekord) ti = <vi1, ... ,vik>

Szemléltetése: a táblázatos forma (reláció, tábla)

R A1 … Ak Aj - attribútumnév

t1 v11 … v1k ti – csak szimbolikusan

… … vezetem be, hogy tudjak

tm vm1 … vmk a sorokra hivatkozni: rowid

## Relációs adatmodell – előfordulás (2/2)

todo

A táblázatos szemléltetésből áttérhetünk a sorok egy másik szemléltetésére:

• t ∈ R sor felfogható függvényként is

• ti : U → értékek, ahol U = {A1, … , Ak}

Ezzel a jelöléssel ti (Aj ) = vij ekvivalens jelöléssel ti [Aj ] vagy ti .Aj (objektum-orientált/metódus típusú jelöléssel)

• Előfordulás: sor-függvények véges halmaza

• Korlátozom a függvényt: X részhalmaza U = {A1, … , An}

Ha X = {Ai1 , … , Ail} akkor t[X] típusa:

t[X] = <Ai1: t(Ai1), … , Ai1: t(Ai1)>

• Függvény szemléltetéssel könnyen tudom képezni a sorok egy részét és így állítom elő a megfelelő sort.

## Relációs adatbázis felépítése

* Az adatbázis tulajdonképpen relációk halmaza.
* a megfelelő relációsémák halmaza adja az adatbázissémát (jelölése vastag szárú R)
* a hozzá tartozó előfordulások az adatbázis-előfordulás
* Előfordulás tartalma: egyes relációk előfordulásai

## Mire kell odafigyelni?

Attribútumok, sorok sorrendje nem számít.

## Kiosztott lap

Tervezéssel később foglalkozunk, a példa hibás, az elnevezésekben cím-cím névütközés. Nem ugyan azt jelentik:

Tankönyv példája, hibás fordítás: title=(film)cím és address=(lak)cím

## Példa megszorításokra - Kulcs (1/3)

* Előző példában: attribútumok aláhúzása mit jelent?
* Filmek: elvárjuk, hogy ne legyen a megengedett előfordulásokban két különböző sor, amelyek megegyeznek cím, év attribútumokon.
* Egyszerű kulcs egy attribútumból áll, de egy kulcs nem feltétlenül áll egy attribútumból, ez az összetett kulcs. Például a Filmek táblában a cím és év együtt alkotják a kulcsot, nem elég a cím, ugyanis van például (King Kong, 1933), (King Kong, 1976) és (King Kong, 2005).
* A kulcsot aláhúzás jelöli: Filmek (cím, év, hossz, …)

## Példa megszorításokra - Kulcs (2/3)

todo

Az attribútumok egy halmaza egy kulcsot alkot egy

relációra nézve, ha a reláció bármely előfordulásában

nincs két olyan sor, amelyek a kulcs összes

attribútumának értékein megegyeznének.

Formális megadása:

R(U), X ⊆ U, U= {A1, … , Ak}, X = {Ai1 , … , Aik}

t ∈ R, t[X] < Ai1 : t(Ai1 ), … , Aik : t(Aik )>

ezzel a jelöléssel mit jelent, hogy X kulcs elvárás?

ha t1 ∈ R, t2 ∈ R és t1[X] = t2[X] akkor t1 = t2

# Keddi előadás 2. rész - Relációs Algebra

Kiosztott papír jobb oldalán konkrét előfordulásokat látunk (táblázatokat)

Hibás séma, a típus oszlopok nem ugyan azt jelölik.

## Milyen típusú feladatokat fogalmazhatunk meg?

todo

Például (lásd az 1. előadáson kiadott lapon a konkrét táblákat)

Legyen adott az alábbi relációs sémák feletti relációk:

Termék (gyártó, modell, típus)

PC (modell, sebesség, memória, merevlemez, ár)

Laptop (modell, sebesség, memória, merevlemez, képernyő, ár)

Nyomtató (modell, színes, típus, ár)

Feladatok (minden kérdést a konkrét táblák alapján természetes módon meg lehet válaszolni, és a köv.héten felírjuk relációs algebrában is)

a.) Melyek azok a PC modellek, amelyek sebessége legalább 3.00?

b.) Mely gyártók készítenek legalább egy gigabájt méretű merevlemezzel rendelkező laptopot?

c.) Adjuk meg a B gyártó által gyártott összes termék modellszámát és árát típustól függetlenül!

!! i.) Melyik gyártó gyártja a leggyorsabb számítógépet (laptop v. PC-t)?

!! k.) Melyek azok a gyártók, akik pontosan három típusú PC-t forgalmaznak?

## Mit nevezünk algebrának?

Nyelv: a kérdés szintaktikai alakja és a kérdés kiértékelése (algoritmus) kiértékelési szemantika

* Algebra műveleteket és atomi operandusokat tartalmaz.
* Relációs algebra: az atomi operandusokon és az algebrai kifejezéseken végzett műveletek alkalmazásával kapott relációkon műveleteket adunk meg, kifejezéseket építünk (a kifejezés felel meg a kérdés szintaktikai alakjának).
* Fontos tehát, hogy minden művelet végeredménye reláció, amelyen további műveletek adhatók meg.
* A relációs algebra atomi operandusai a következők:
  + a relációkhoz tartozó változók,
  + konstansok, amelyek véges relációt fejeznek ki.

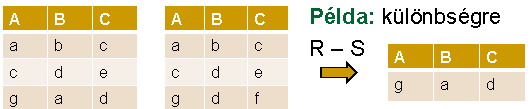
## Halmazműveletek (jelölése a szokásos)

todo

Reláció előfordulás véges sok sorból álló halmaz. Így értelmezhetők a szokásos halmazműveletek: az unió (az eredmény halmaz, csak egyszer szerepel egy sor) értelmezhető a metszet és a különbség. Milyen művelet van még halmazokon? Értelmezhető-e relációkon?

R, S és azonos típusú, R ∪ S és R – S típusa ugyanez R ∪ S := {t | t∈R ∨ t∈S}, R – S := { t | t ∈ R ∧ t ∉ S}

Az alapműveletekhez az unió és különbség tartozik, metszet műveletet származtatjuk R ∩ S = R – (R – S)



Multihalmaz: Egy sor többször is előfordulhat a táblában.

Metszet, szimmetrikus differencia értelmezve van relációkra, kifejezhető alapműveletekkel.

## Vetítés (project, jelölése pí)

todo

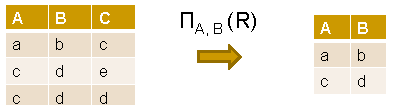
Vetítés (projekció). Adott relációt vetít le az alsó indexben szereplő attribútumokra. (attribútumok számát csökkentik)

Πlista(R) ahol lista: {Ai1 , … , Ail} R-sémájában levő attribútumok egy részhalmazának felsorolása eredmény típusa <Ai1: értéktípusi1, … , Ail:értéktípusil>

Πlista(R) := { t.Ai1 , t.Ai2 , … , t.Ail] | t∈R}

Reláció soraiból kiválasztja az attribútumoknak megfelelő Ai1 , … , Ail -n előforduló értékeket, ha többször előfordul akkor a duplikátumokat kiszűrjük (hogy halmazt kapjunk)

Példa:



c, d kétszer szerepelne, ezért az egyiket kihagyja.

SQL-ben SELECT DISTINCT kulcsszóval külön kell kérni a duplicate törlést.

## Kiválasztás (select, jelölése: kis-szigma)

todo

Kiválasztás (szűrés). Kiválasztja az argumentumban szereplő reláció azon sorait, amelyek eleget tesznek az alsó indexben szereplő feltételnek.

σ\_Feltétel(R) és R sémája megegyezik

σ\_Feltétel(R) := { t | t∈R és t kielégíti az F feltételt}

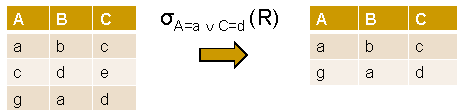
R(A1, …, An) séma feletti reláció esetén a σF kiválasztás

F feltétele a következőképpen épül fel:

Elemi feltétel: Ai θ Aj, Ai θ c, ahol c konstans, θ pedig =, ≠,<, >, ≤, ≥

Összetett feltétel: ha B1, B2 feltételek, akkor ¬ B1, B1∧ B2, B1∨ B2 és zárójelezésekkel is feltételek

Példa:



SQL-ben ez a WHERE feltétel.

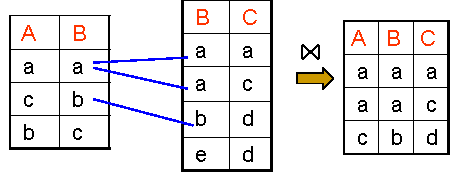
## Természetes összekapcsolás (natural join) (jelölése: "csokornyakkendő")

todo

Szorzás jellegű műveletek (attribútumok számát növeli) többféle lehetőség, amelyekből csak egyik alapművelet:

Angolul: Natural Join (jelölése: "csokornyakkendő")

Természetes összekapcsolás: közös attribútum-nevekre épül. R ⋈ S azon sorpárokat tartalmazza R-ből illetve S-ből, amelyek R és S azonos attribútumain megegyeznek.



Természetes összekapcsolás:

Legyen R(A1,…,Ak,B1,…,Bn), illetve S(B1,…,Bn,C1,…,Cm)

R ⋈ S típusa (A1,…,Ak,B1,…,Bn,C1,…,Cm) vagyis a két attribútumhalmaz uniója

R ⋈ S elemei v ∈ R ⋈ S

R ⋈ S = { v | létezik t ∈ R és s ∈ S és

t[B1,…,Bn] = s[B1,…,Bn] és

v[A1,…,Ak] = t[A1,…,Ak] és

v[B1,…,Bn] = t[B1,…,Bn] és

v[C1,…,Cm] = s[C1,…,Cm] }

Példákban: két azonos nevű attribútumot úgy tekintünk, hogy ugyanazt jelenti és a közös érték alapján főzzük össze a sorokat.

Milyen problémák lehetnek?

Filmek adatbázisban ugyanarra a tulajdonságra más névvel hivatkozunk: Filmek.év és SzerepelBenne.filmÉv, illetve FilmSzínész.név és SzerepelBenne.színészNév

Termékek adatbázisban pedig ugyanaz az azonosító mást jelent: Termék.típus más, mint Nyomtató.típus

Így a Filmek és a Termékek adatbázisokban ahhoz, hogy jól mőködjön a természetes összekapcsolás szükségünk van egy technikai mőveletre: az átnevezésre.

## Átnevezés

Jövő héten lesz.

Előadás vége.

Járhatok mindkét előadásra. Kis háromszög a slide-ok négy lapján jelentve. Plusz pontot nem kapok érte, de ha jelentek még hibákat összejön.

A sor jele azért , mert (n-es)