

Budapesti Corvinus Egyetem, Számítástudományi Tanszék

Adatbázis rendszerek előadás jegyzet 2017

Kardkovács Zsolt előadása, Kerepes Tamás jegyzeteivel



ADATBÁZISOK

Kardkovács Zsolt Tivadar

Adatbázisok

Relációs adatmodell

*„A matematikus egy olyan készülék,
amely egy kávéból is elméletet csinál.”
(Erdős Pál)*



Noha a számítógép első felhasználása a „számolás” volt, manapság gyakrabban használjuk adatok tárolására, azok bevitelére, módosítására, feldolgozására, mint hagyományos értelemben vett „számolásra”.

Ha már adatokat tárolunk, akkor azokat a számítógép azon részén, amelyen az adatok megmaradnak akkor is, ha a számítógépet kikapcsoljuk, vagy hiba folytán leáll (az adatok ne vesszenek el). Ez manapság leginkább a merevlemez (hard disk, hard drive), lehet ez szalagos mentőegység, de megjelennek más fizikai tárolóegységek és módszerek is – pl. flash diszkek, stb.

Ezek az adatok – főként ha csak magunk használjuk őket – nem feltétlenül, sőt nem is elsősorban adatbázisokban tárolódnak. Ezek leginkább fájlok (állományok) formájában tárolódnak a merevlemezeinken. Ilyen pl. a sok-sok Word Dokumentum állomány, a sok Excell file, vagy akár a sok TEXT-file, amelyet pl. Notepad vagy Wordpad program segítségével készítettünk el, majd lementettük őket a merevlemezre.


Az imént említett file-ok nagy száma miatt szükség van valamiféle rendszerezésre, és ezért a rengeteg fájlt könyvtárakba és alkönyvtárakba csoportosítjuk. Így pl. a laptopon jelenleg 670.000 állományt tárolok. A fájlok nagy száma azonban még mindig nem indokolja az adatbázisok bevezetését.

Adatbázis ott kell, ahol a nagy adatmennyiség mellett az adatok jelentősen összefüggnek egymással, gyakran több felhasználó használja ezeket az adatokat, szükség van azok konzisztens kezelésére. Pl. egy webáruház termékeket mutat a potenciális vásárlók számára, de a raktáron ezen termékekből véges számú van. Ezért nem megengedhető, hogy a vevők több terméket vásároljanak meg, mint amennyi a raktárkészlet. Ezt a feladatot olyan alkalmazások valósítják meg, amelyek mögött adatbázis tárolja a raktárkészletet, a rendeléseket, stb. Vagy pl. a Neptun rendszer nem elsősorban egy internetes program, hanem legfőképp egy adatbázis sok-sok egymással összefüggő adattal, amelyet aztán Webes alkalmazások használnak. Ha nem adatbázis lenne, hanem csak kutyaközséges text-állományok, akkor nem tudnánk biztosítani az egyidejű konzisztens hozzáférést sok felhasználó által.

Adatbázisnak a valós világ egy részhalmazának leírásához használt adatok összefüggő, rendszerezett halmazát nevezzük. – By dr. Gajdos Sándor: Adatbázisok, BME Villamosmérnöki és Informatikai Kar.

Többféle gyártó készített az évtizedek során különböző típusú adatbázis-megoldásokat. Mindegyikre jellemző azonban, hogy létezik maga az adatbázis (database) és az adatbáziskezelő rendszer (Database Management System).

Az adatbáziskezelő rendszer egy szoftver. Megállítható, majd újraindítható, és „generikus”. Az adatok „egyediek” és az adatbázisban tartósan tárolódnak. „Cégspecifikusak” - pl. két versenytárs nagyban ugyanolyan adatbáziskezelő rendszert használhat – mondjuk Oracle RDBMS-t, annak is a 12cR2 változatát. De az adataiknak nemcsak a tartalma, hanem még a szerkezete is különbözik egymástól. (Ez egyébként egy Excell program és egy Excell táblázat esetén is így van.)



CORVINUS
UNIVERSITY OF
BUDAPEST

ADATBÁZISOK

Adatmodell

- Adatmodell:
 - „**logikai struktúra** és azon értelmezett **kényszerek és műveletek** összessége”

Minden adatbázisra jellemző műveletek

- (érték) Beszúrás
- (érték) Törlés
- (érték) Módosítás (= Törlés + Beszúrás)
- (jellemző) Átnevezés
- (jellemző) Törlés
- (jellemző) Létrehozás
- (adatszerkezet) Létrehozás
- (adatszerkezet) Törlés
- ... stb.

Aki figyelt:
logikai művelet ≠ fizikai művelet

Amikor egy adatbázist létrehozunk, a cél az, hogy benne a való világ adatait tároljuk, hogy belőle később információkat nyerhessünk, ahelyett, hogy a valóságból kelljen ugyanazokat az információkat megszerezni.

Általában nincsen mód (nem érdemes) egy problémakörrel kapcsolatos minden adatot tárolni. Ehelyett azoknak egy szűk körét választjuk ki és kezeljük.

A kiválasztásnál klasszikus modellezési szempontok érvényesülnek: a szempontunkból fontosnak tartott információkat tároljuk, a többit elhanyagoljuk. Jellemző alatt egyaránt értünk tulajdonságokat és kapcsolatokat. Így az adatbázis a világ egy darabjának egy leegyszerűsített képét képes visszaadni. (by Gajdos Sándor)

Kényszerekről később lesz szó, de pl. ha a népességnyilvántartó rendszer adatait tervezzük, és 1-1 lakosnak életkora van, akkor az nem lehet negatív szám. Persze ez akkor igaz, ha úgy döntöttünk, hogy 1-1 állampolgárunk a születésekor kerül be ebbe a rendszerbe, és nem a pl. fogamzáskor.



ADATBÁZISOK

Milyen adatmodellek ismertek?

Milyen adatmodellek ismertek?

- Hálós adatmodell
- Hierarchikus adatmodell
- **Relációs adatmodell**
- Objektumorientált adatmodell
- Logikai (vagy deduktív) adatmodell
- Objektumrelációs adatmodell
- Deduktív objektumorientált adatmodell
- Kulcs-érték adatmodell
- ...

Kereskedelmi adatbázisok döntő többsége (98%+) relációs

Webes adatbázisok többsége nem (NoSQL)

Termékek (DBMS!):

- Oracle, DB/2, MS SQL Server, Sybase ASE, Informix, MySQL, Postgres...

Jelenleg a messze legnagyobb piaci részesedése a relációs adatbáziskezelőknek van.

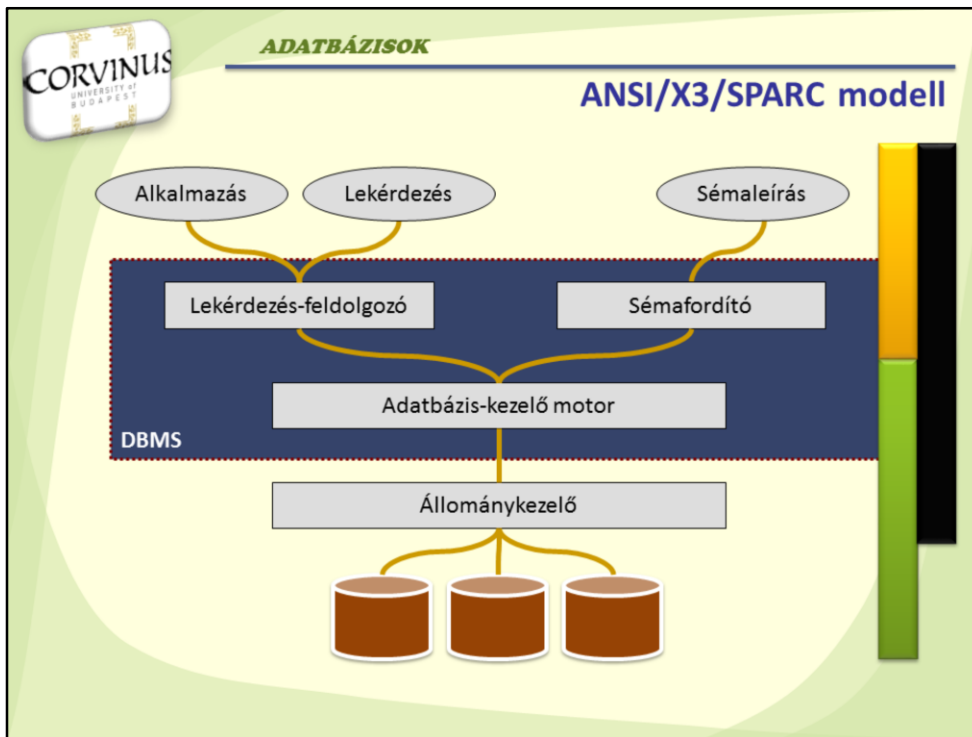
Mivel a programok írásában (programozásban) az objektumorientáltság a legelterjedtebb (ez a divatos), így elméleti szakértők szerint az objektumorientált adatbázisoknak kéne felváltaniuk a relációsakat, de ez a folyamat nem tetten érhető – mintha nem történne meg, vagy csak egyelőre nem történik.

A relációsak között a jelenlegi piacvezető az Oracle RDBMS (Relational Data Base Management System). Fontos még a „nagy és fizetősek” között az MS SQL Server, valamint a „kicsik” között a MySQL és talán a Postgres. A verseny éles, és változik az is, melyikek a fontosak, és melyikek nem annyira.

A magyar piacon aránytalanul nagy túlsúlya van az Oracle adatbáziskezelő rendszernek, a DB/2 pedig ritkaságszámba megy. A nemzetközi versenyben is első az Oracle, de ott közelebb vannak hozzá a versenytársai: az MS SQL Server és a DB/2.

Az Informix, a Sybase ASE, a Teradata és sok másik mintha lehulló ágon lennének.

A relációs adatbázis kezelő rendszerek többsége pusztán szoftver, és általános (közönséges) számítógépeken működnek (többnyire Linuxon) – tehát nem erre a célra megépített hardveres megoldás. Van azonban néhány olyan megoldás amely nemcsak szoftver, hanem célhardver is (ún. „database computer”. Ilyen pl. a Teradata és az Oracle Exadata és Oracle Supercluster megoldások. („Engineered solutions”)



Az első adatbázisok már az 1960-as években megjelentek. Elsőként a hierarchikus és a hálós adatbáziskezelők születtek meg (pl. az IBM-nél). A hálós adatbáziskezeléshez még egy szabvány is készült. A CODASYL (Conference on Data Systems Language) 1959-ben alakult és két nagy dolgot alkotott:

1. a COBOL programozási nyelvet (1959)
2. a CODASYL adatmodell (1969) – a hálós adatbáziskezelés „szabványa”

1970: Edgar Frank (Ted) Codd cikke: „A Relational model for large shared data banks”

Mai napig is Codd számít a relációs adatbáziskezelés atyjának.

Az ANSI SPARC modell 1975-ös: Standards Planning and Requirement Committee. Nem lett teljes mértékű elfogadott szabvány, csupán az ötleteiből merítettek a különböző gyártók.

Az ANSI SPARC modell három rétegből áll:

1. a legfelső szint (External level): felhasználói nézet, amelyben a felhasználó csak azokat az adatokat látja, amelyek őt érdeklik, illetve amelyekhez hozzáférése van.
2. a konceptuális szint: az adatbázis összes adatának a leírása, és azok összefüggéseinek a leírása. A fizikai tárolás módjával ez még nem foglalkozik.
3. belső szint (Internal level): hogyan tárolódnak fizikailag az adatok.

Eszerint létezik: „External Schema”, „Conceptual Schema”, „Internal Schema”.



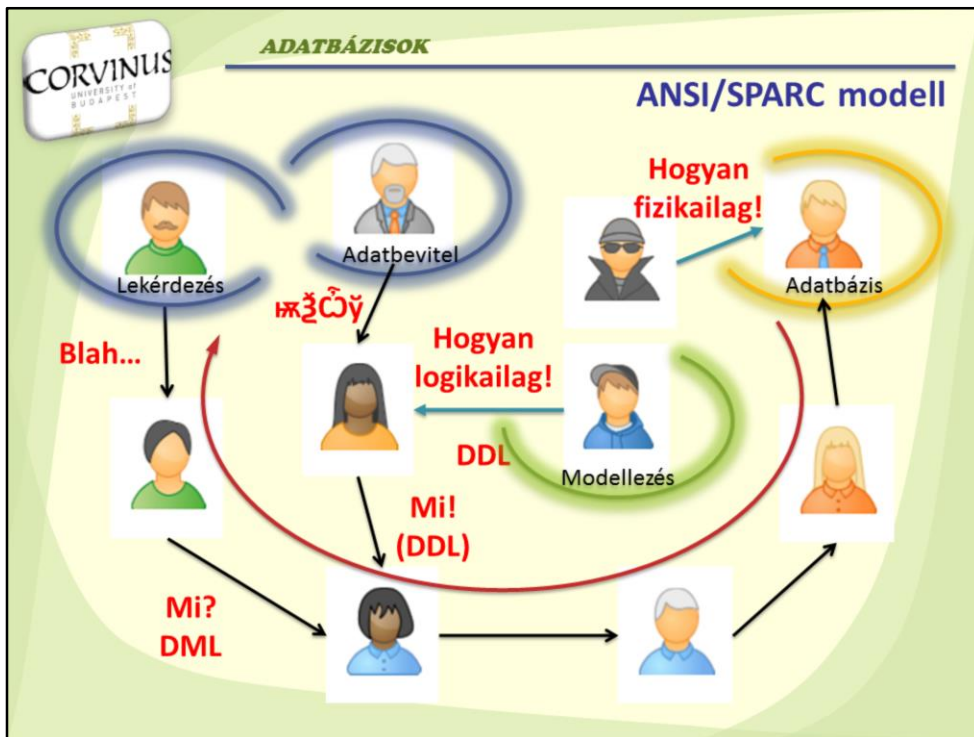
Az alkalmazások (programok) végfelhasználói is, meg a fejlesztők is csak azt látják, amihez joguk van (external level). Így egyszerűnek látják az adatok szerkezetét.

Az adatbázis rendszergazdái (DBA) és a logikai tervezői (modellezői) látják a teljes adatmodellt (conceptual schema).

Az adatbázis adatainak a fizikai tárolási módja, az ún. „internal schema” ismerete még a rendszergazdák számára sem nélkülözhetetlen. Egyes rendszereknél ezt el sem árulja nekik az adatbáziskezelő szoftver gyártója, mások dokumentálják ezt, de nagy kérdés, hogy szükséges-e ezt tudnia az adatbázis rendszergazdájának. Ilyen kérdések pl.:

- a) Hogyan tárolják ezek a rendszerek a szöveges adatokat
- b) Hogyan tárolják a számokat
- c) Hogyan tárolják a dátum típusú adatokat,
- d) A merevlemezen a rekordok milyen logika szerint tárolódnak: 1-1 Byte mit jelent/jelez.
- e) Stb., stb.


Vannak azért olyan „internal schema” részletek, amelyekről tudnia, sőt döntenie kell a DBA-nak: ilyen részletek pl. hogy melyik lemezen, melyik könyvtárakban milyen nevű és méretű állományok tárolják az adatbázis adatait, stb.



Egy adatbázis használata során különböző szereplőket látunk. Ezek:

1. Végfelhasználó („End User”, aki egy mások által megírt alkalmazást használ és azon keresztül visz be adatot. Pl. egy banki ügyintéző, vagy egy éppen repülőjegyet vásárló ügyfél.
2. „Lekérdező”, aki az adatbázis adatait olvassa. Pl. adatbányászat, döntéstámogatás, felhasználók informálása. Neki ismernie kell azt a lekérdező „nyelvet”, amely ezt lehetővé teszi.
3. Elemző/tervező: leginkább ugyanaz a személy. Ő megérti az üzleti folyamatokat, azokat modellezi és ezekhez illő adatszerkezeteket tervez/hoz létre.
4. Fejlesztő: olyan alkalmazásokat (programokat) ír, amelyeket a végfelhasználó vagy a lekérdező használnak. Ezek a programok gyakran erősen adat-orientáltak. Gyakran az elsődleges szerepük az adatok bevétele vagy megjelenítése. Ritkábban persze komoly feldolgozások is történnek, ahol esetleg az adat tárolása, visszakeresése nem a legnehezebb/legfontosabb feladat.
5. Adatbázis rendszergazda:
 1. Létrehozza a fizikai adatbázist
 2. Biztosítja annak elérhetőségét, működését
 3. Felhasználókat és jogosultságokat kezel
 4. Menti és szükség esetén helyreállítja az adatbázist.
 5. Hangolja a rendszer működését
 6. Együttműködik az oprendszer, a hálózat és a diszkrendszer rendszergazdáival

Egy adatbázist nagyon sok különféle program használhat. Ezek mind egy „nyelvet” használnak az adatok eléréséhez. Ezt API-nak nevezik az informatikusok. Ennek az API-nak részei a DDL-ek és a DML-ek a fenti ábrán.



CORVINUS
UNIVERSITY OF
BUDAPEST

ADATBÁZISOK

Relációs adatmodell (1. rész)

Logikai struktúrák

- **Attribútum**
 - „egyedek azonos minőségű jellemzőinek halmaza”
 - -> attribútumérték (érték): ezen halmaz egy eleme
 - Jelölés: A1, A2, ..., pl. Lakcím, Név, ...
- **Séma**
 - „attribútumok névvel nevezett viszonya”
 - Jelölés: R(A1, A2, A3, ...), pl. Ember(Név, Lakcím)
- **Reláció (matematikai fogalom!)**
 - „jellemzők Descartes-szorzatának **részhalmaza**”
 - Jelölés: r(R) -> az R sémára illeszkedő reláció.

Mindig feltételezhetjük, hogy egy attribútum véges halmaz

Hány séma lehet?

Hány reláció lehet?


A relációs adatmodell mögött a halmazelméleti relációk elmélete áll.

Attribútum: pl. Lakcím, Név.

Séma: Ember(Név, Lakcím). Ha pl. 3 attribútum van, akkor $2^3 - 1$ séma lehetséges.

Matematikai, elméleti definíció: Attribútum-halmazok Descartes szorzatának részhalmazát relációnak nevezzük. Ha van 4-féle Név és 3-féle Lakcím, akkor a Descartes-szorzatban $4 \cdot 3 = 12$ elem van. Ezek bármely részhalmazát relációnak nevezzük.

Relációs séma: gyakran magára a relációra (tehát az „adatokra” nincs szükségünk, csak arra az információra, hogy melyik relációban milyen attribútumok találhatóak. Ezt relációs sémának nevezzük.



ADATBÁZISOK

Reláció (példa)

Matematikai ábrázolás

▪ Hallgatók{
 < Gipsz Jakab ; ABC123 ; 4632031 ; Sóház u. 13. ; 22 ; 3,52 > ,
 < Ency Klopédia ; ABC124 ; 4632045 ; Sóház u. 13. ; 23 ; 4,1 > ,
 ...)

Rendezett halmaz?

Vizuális (táblázatos) ábrázolás

Hallgatók	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Ency Klopédia	ABC124	4632045	Sóház u. 13.	23	4,1
	Kövér Margó	CBA123	4632027	Sóház u. 13.	22	3,78
	Sánta Kutya	ACB123	4632976	Sóház u. 13.	24	4,8

A relációk matematikai ábrázolása is lehetséges, de sokkal könnyebb táblázatosan ábrázolni őket.

A reláció neve: Hallgatók.

A táblázat oszlopai az attribútumok. Ezeknek értékhalmaza van. Pl. 4 különböző Név, 3 különböző Kor, 4 különböző Átlag attribútumérték látható itt.

A Descartes szorzat ezeknek mindenféle lehetséges kombinációja lenne. De mint látjuk, a 4 különböző név * 3 különböző Kor 12 kombinációt adna, de mi csak ezt részhalmazt választottuk ki.

Végül a relációnknak csak 4 eleme van.

A sorok (n-esek, n-tuples) sorrendjének nincs jelentősége.

Matematikai, elméleti síkon az oszlopok sorrendje sem számítana, de később a lekérdező nyelvben esetleg az oszlop sorszámával hivatkozunk majd rá - ezért ez a gyakorlatban számít.

A reláció nem tartalmazhat két azonos sort.

Az oszlopoknak egyértelmű nevük van.

A relációs adatbázisokban ezeket a „relációkat” **tábláknak, vagy relációs tábláknak** nevezzük.

Súlyos tévedések forrása lehet, ha csupán az oszlopok nevére hagyatkozunk, és nem definiáljuk precízen a jelentésüket. Pl. az Átlag lehetne akár „Átlag életkor” is.

Matematikai és vizuális megfeleltetések

Matematikai fogalom

- Reláció
- Attribútum
- Reláció eleme / ennes / tuple
- Attribútumérték
- Séma

Vizuális (táblázatos) fogalom

- Táblázat
- Oszlop
- Sor
- Cella
- Táblázat szerkezete, fejléce

Különbségek?
(több is van...)

Különbségek:

A matematikailag definiált reláció nem tartalmazhat két azonos sort. A relációs adatbázis táblájában ez mégis tárolható – de ez bajt okoz majd, tehát kerülni kell

Az attribútumok sorrendje elméleti síkon irreleváns. A táblázatos formában, tehát a táblákban nyilván van jelentősége.

Relációs adatmodell (2. rész)

Kényszerek

▪ Szuperkulcs:

- „olyan attribútumhalmaz, amelynek értékei egyértelműen meghatározzák, hogy melyik elemről beszélünk”

▪ Kulcs:

- „olyan szuperkulcs, amelynek egyetlen valódi részhalmaza sem szuperkulcs”

▪ Funkcionális függőség:

- „Amennyiben egy A attribútumhalmaz értékei megegyeznek egy reláció elemein, akkor szükségszerűen B attribútumhalmaz értékei is megegyeznek, akkor azt mondjuk, hogy B függ A-tól; ezt $A \rightarrow B$ formában jelezzük.”

▪ Egyediségi kényszer:

- „olyan attribútumhalmaz, amelynek értékei nem ismétlődnek a relációban”

Miért fontosak ezek a gyakorlatban?

PL az EMBER entitás (reláció), azaz tábla egyik kulcsa a (Név, Szül.dátum, Anyjaneve) attribútumhármass.

Ugyanannak a táblának egy másik kulcsa a személyi szám.

Ha pl. egy táblában tároljuk a Város attribútumot is, meg az Országot is, és két különböző Országban nem lehet azonos nevű Város (pl. Budapest nevű város egyértelműen a mi Budapestünket jelenti, akkor mindig, amikor két sorban ugyanaz a város (tehát Budapest), akkor ugyanaz az ország is (tehát Magyarország). Város \rightarrow Ország funkcionális függőség áll fenn.

Igaz vagy hamis? (Indoklással!)

- Minden relációnak van superkulcsa ($\forall R \exists A: A \rightarrow R$).
- Ha A egyedi, akkor A superkulcs.
- Ha A superkulcs, akkor A egyedi a relációban.
- Az egyediség és a superkulcs tulajdonság azonos.
- Minden relációnak pontosan egy kulcsa van.
- Van olyan reláció, amelynek több olyan kulcsa is van, amelyek között nincs egyforma attribútum.
- A függőségek a relációk alapján is felderíthetők.



$r(R)$ reláció, A és B
attribútumhalmaz

Extra feladat

- Mekkora elemszámú reláció kell ahhoz, hogy kizárólag az általunk megadott függőségek legyenek igazak egy reláción?
- Kulcsszó: CODD

Relációs adatmodell (3. rész)

Műveletek

- A szokásos műveleteken túl... Matematikai értelemben mit változtatnak ezek?
- Halmaz(!)műveletek mint alpműveletek
 - Vetítés vagy projekció – általában π jelöli
 - Kiválasztás, szűrés vagy szelekció – általában σ jelöli
 - Descartes-szorzat (kartéziánus) – általában \times jelöli
 - Unió – általában \cup jelöli
 - Különbség – általában \setminus jelöli
- Származtatott műveletek (például)
 - Metszet, hányados
 - Természetes illesztés vagy join (és a részleges illesztések)
 - Theta illesztés...

A beszűrés pl. származtatott művelet?

Matematikai ábrázolás

- Unió = Hallgatók1 \cup Hallgatók2

Hallgatók1	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Ency Klopédia	ABC124	4632045	Sóház u. 13.	23	4,1

Hallgatók2	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Sánta Kutya	ACB123	4632976	Sóház u. 13.	24	4,8

Unió	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Ency Klopédia	ABC124	4632045	Sóház u. 13.	23	4,1
	Sánta Kutya	ACB123	4632976	Sóház u. 13.	24	4,8

Matematikai ábrázolás

- $\text{Különbség} = \text{Hallgatók1} \setminus \text{Hallgatók2}$

Hallgatók1	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Ency Klopédia	ABC124	4632045	Sóház u. 13.	23	4,1

Hallgatók2	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Sánta Kutya	ACB123	4632976	Sóház u. 13.	24	4,8

Különbség	Név	Neptun	Telefon	Cím	Kor	Átlag
	Ency Klopédia	ABC124	4632045	Sóház u. 13.	23	4,1

Descartes-szorzat

Matematikai ábrázolás

- Szorzat = Hallgatók1 x Hallgatók2

Mekkora lesz az eredményhalmaz?

Hallgatók1	Név
	Gipsz Jakab
	Ency Klopédia

Hallgatók2	Név
	Gipsz Jakab
	Sánta Kutya

Szorzat	Név	Név
	Gipsz Jakab	Gipsz Jakab
	Gipsz Jakab	Sánta Kutya
	Ency Klopédia	Gipsz Jakab
	Ency Klopédia	Sánta Kutya

Érdemes átnevezni!

Matematikai ábrázolás

- Kiválasztás = $\sigma(\text{Hallgatók} \mid \text{Neptun} = \text{'ABC123'})$

Hallgatók	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Sánta Kutya	ACB123	4632976	Sóház u. 13.	24	4,8

Kiválasztás	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52

Matematikai ábrázolás

- $Vetítés = \pi[\text{Neptun}, \text{Cím}](\text{Hallgatók})$

Hallgatók	Név	Neptun	Telefon	Cím	Kor	Átlag
	Gipsz Jakab	ABC123	4632031	Sóház u. 13.	22	3,52
	Hibás Elemér	ABC123	4632976	Sóház u. 13.	24	4,8

Vetítés	Neptun	Cím
	ABC123	Sóház u. 13.

Mi történt?

Származtatott műveletek

- Metszet = $A \cap B = A \setminus (A \setminus B) = B \setminus (B \setminus A)$
- Illesztés = $A \bowtie B = \pi [\dots] (\sigma (A \times B \mid \dots))$
- Hányados = $A / B = \pi [\dots] (A) \setminus \pi [\dots] ((\pi [\dots] (A) \times B) \setminus A)$

Hallgatók	Név	Neptun	Átlag
	Gipsz Jakab	ABC123	3,52
	Sánta Róka	ACB123	4,8

Jegyek	Tárgy	Neptun	Jegy
	Adatbázisok	ABC123	4
	Adatbázisok	ACB123	5

Illesztés	Név	Neptun	Átlag	Tárgy	Jegy
	Gipsz Jakab	ABC123	3,52	Adatbázisok	4
	Sánta Róka	ACB123	4,8	Adatbázisok	5