

Adatbázis-kezelés

Takács Gábor

mérnökinformatikus, okl. mérnöktanár
Informatika Tanszék, mesteroktató

takacsg@sze.hu

<http://takacsg.hu/>



Adatbázis-kezelés alapok



Miért nem jó?

- Redundáns (ismétlődő) adatok
- Adattípusok konzisztenciáját nem biztosítja
- Nem kereshető, nehezen átlátható
- Nem biztonságos
- Nincs verziókövetés

MI A MEGOLDÁS?

Adattárolás evolúciója

- Klasszikus többrétegű



Felhasználói felület

Üzleti logika

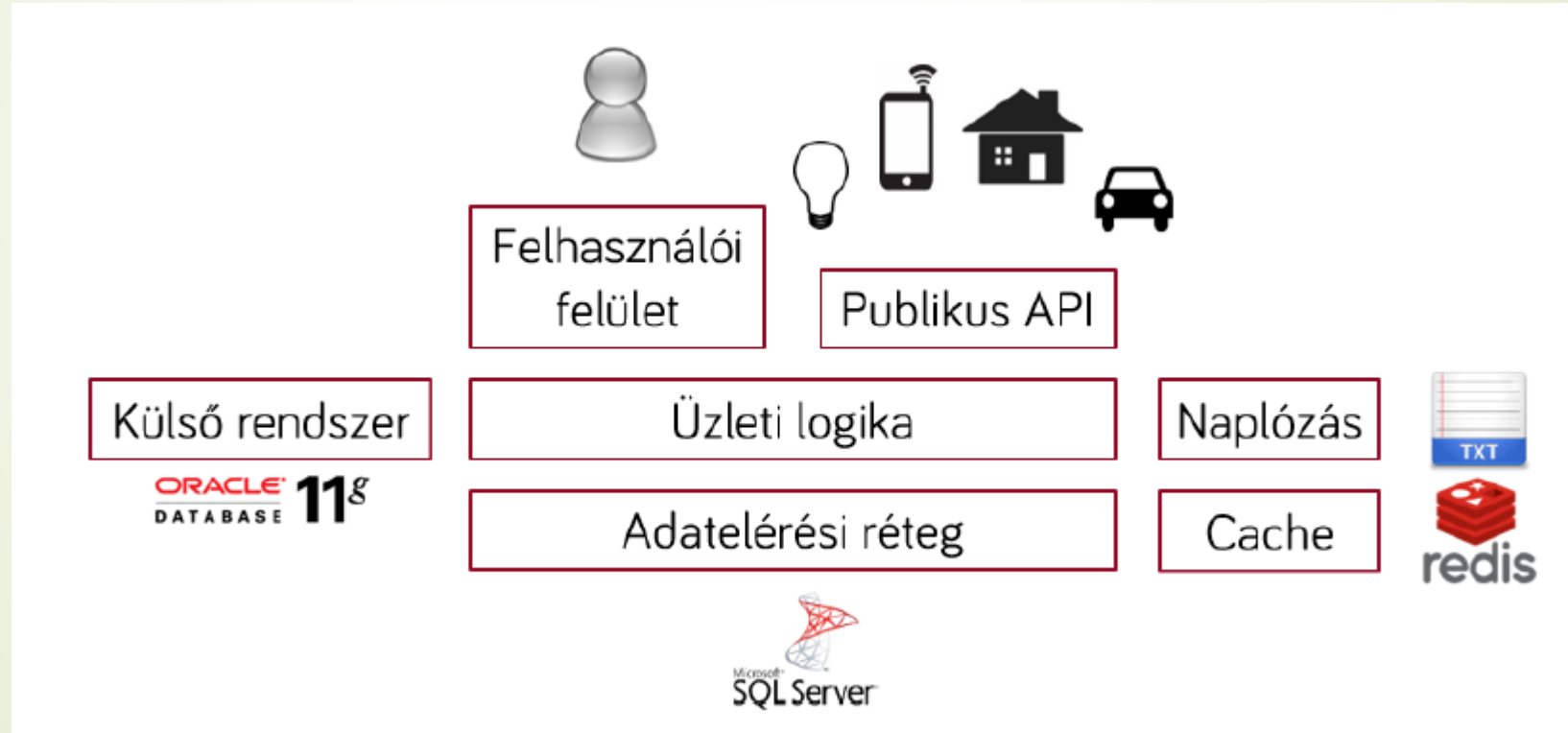
Adatelérési réteg

Adatbázis

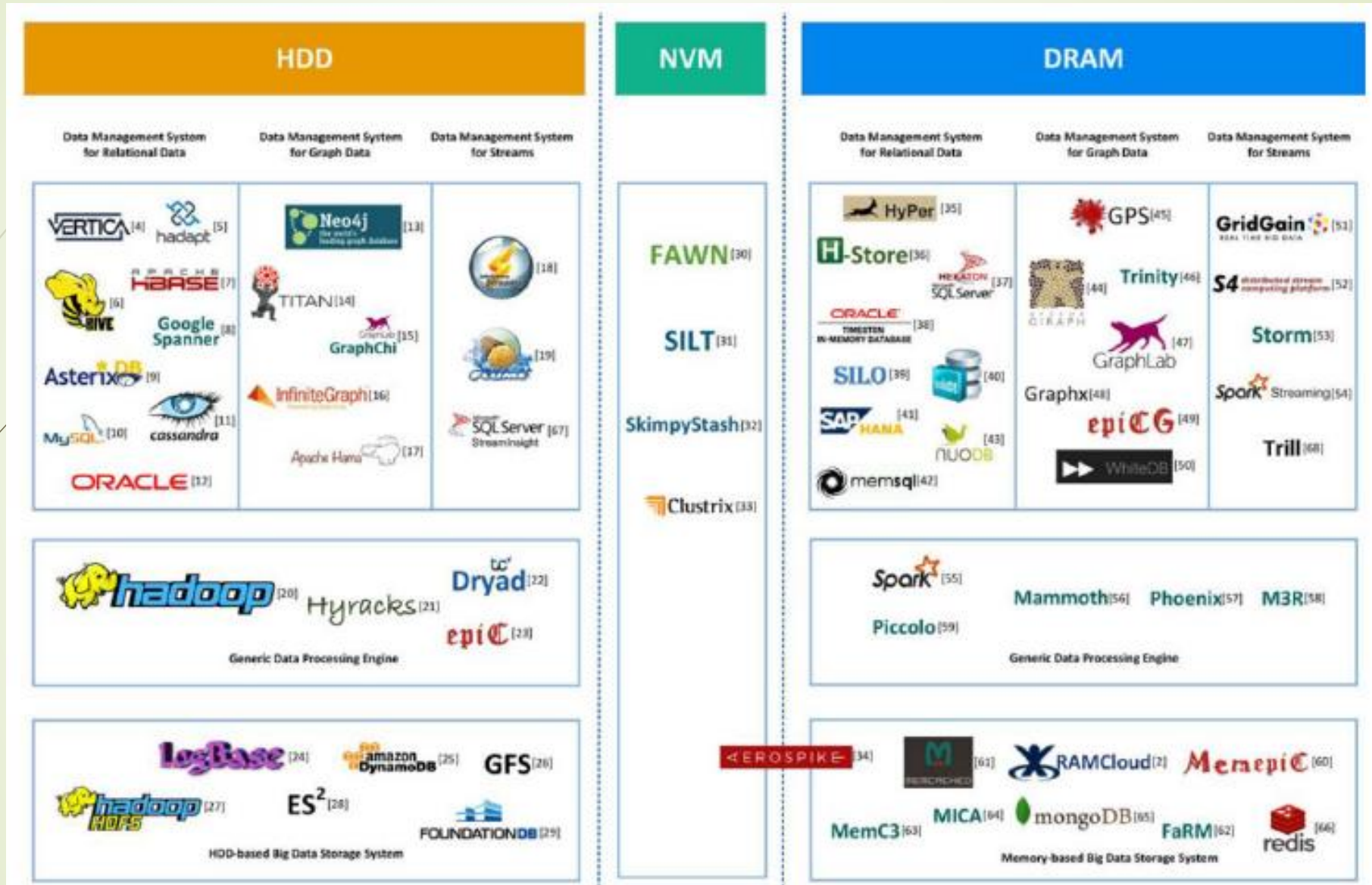
MI A MEGOLDÁS?

Adattárolás evolúciója

- Mini alkalmazások



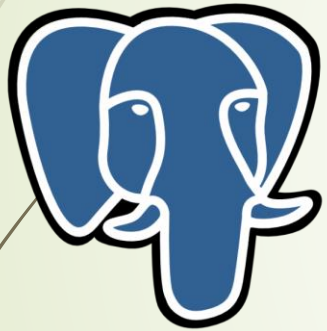
MI A MEGOLDÁS?



HaoZhang; Gang Chen; BengChinOoi; Kian-LeeTan; MeihuiZhang "In-Memory Big Data Management and Processing: A Survey", *IEEE Transactions on Knowledge and Data Engineering*, vol.27, no.7

MI A MEGOLDÁS?

Relációs adatbázis-kezelő használata



PostgreSQL



Microsoft®
SQL Server®



- Táblák
- Mezők
- Rekordok
- Kapcsolatok
- Kulcsok



Adatmodellek:

- **Relációs adatmodell:**

A relációs adatmodellben az egyedet egy táblázattal adjuk meg.

- A táblázat oszlopai az egyedre jellemző tulajdonságok.

- A táblázat sorai a REKORD-ok, az egyed értékei (egyedtípus előfordulásai).

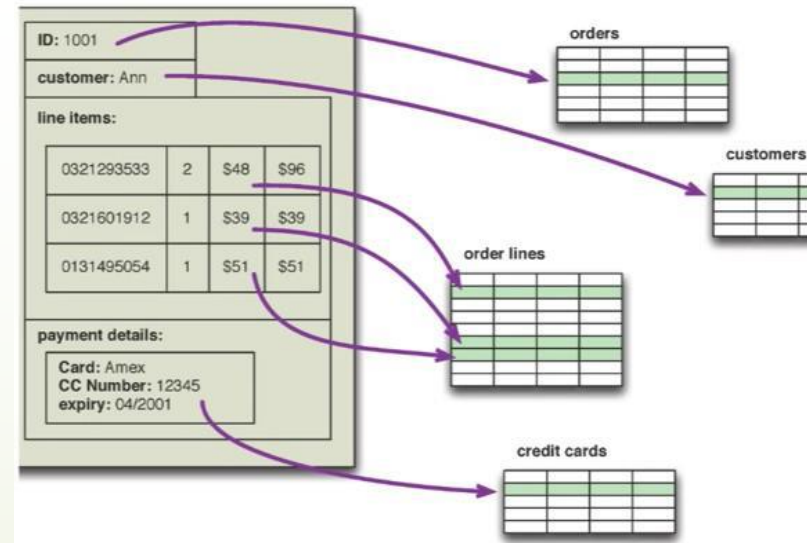
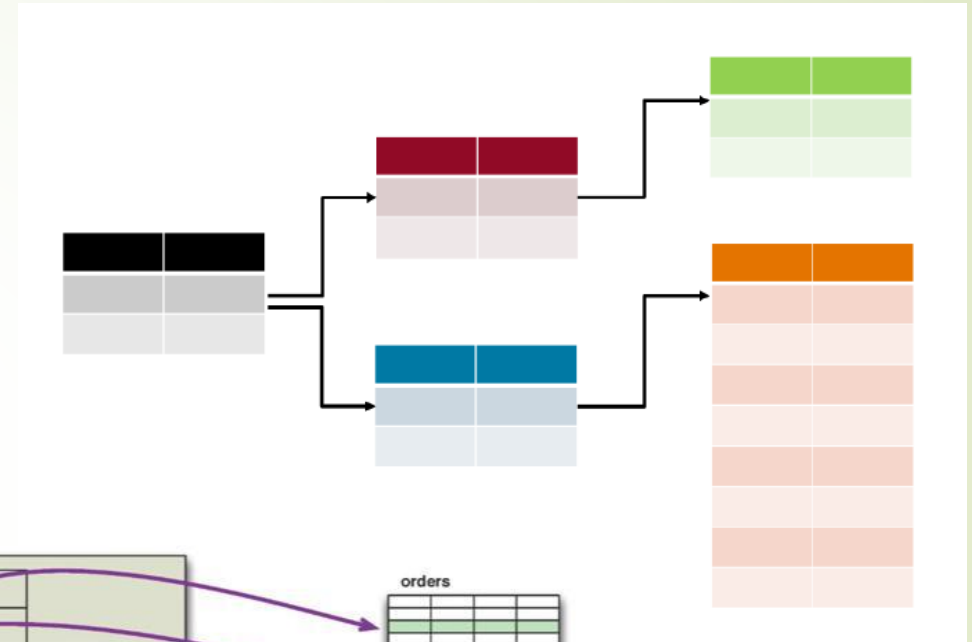
A táblák közötti kapcsolatokat közös tulajdonságokkal, indexeken keresztül valósítjuk meg.

- **Objektum-relációs modell:**

A modell **objektumközpontú elemekkel, tulajdonságokkal (osztályok, egységbezárás, öröklés stb.) bővíti ki a relációs modellt.** Az osztály a relációnak felel meg. Az osztályok példányai az objektumok. Az objektumok közötti kapcsolatokat gyakran asszociációknak nevezzük. Lehetővé teszi összetett adattípusok használatát.

Relációs adatbázisok használata

- Egyedek és köztük levő relációk (kapcsolatok)
- Normalizált séma (általában 3NF)
- Beszúrás, módosítás kevés rekordot érint
 - Normalizálás miatt
 - Erre optimalizált működés
- Adatmodell jól ismert célokat szolgál
 - Előre ismert lekérdezések
 - Erre optimalizált séma



RDBMS Terminológia

Tábla

Sorok (Rekordok)

Oszlopok

Mezők

| id | nev | varos |
|----|---------------|-----------|
| 1. | Kiss István | Győr |
| 2. | Nagy Viktória | Budapest |
| 3. | Kovács Mihály | Szekszárd |
| 4. | Szabó János | Ács |

Fogalmak: egyed, tulajdonság, kapcsolat

- **Egyed:** (Entity) Minden olyan dolog (objektum), ami minden más dologtól (objektumtól) megkülönböztethető. Pl.: Személy, autó, stb.
- **Tulajdonság:** (Attribute) Az egyedeket tulajdonságokkal (attribútumokkal) írjuk le.
Az autó egyedtípus esetén pl.: Típus, rendszám, szín, alvázszám, stb.
- **Kapcsolat:** (Relationship) Az egyedhalmazok közötti viszony fogalmi tükörképe.

1:1, 1:N, (M:1), M:N

KAPCSOLAT

- **Egy-az-egyhez kapcsolat (1 : 1) Pl.: Motorszám - alvázszám**

(A - B) kapcsolat esetén az A egyedhalmaz (szülő) minden egyes eleméhez legfeljebb egy elem tartozhat a B egyedhalmazban (gyermek), és a B egyedhalmaz minden egyes eleméhez is csak legfeljebb egy elem tartozhat az A egyedhalmazban. Egy-az-Egyhez típusú kapcsolatot általában az alábbi esetekben használunk:

- Egy sok elemből álló egyedhalmazt több kisebb, könnyebben kezelhető egyedhalmazra kívánunk felosztani.
- Egy egyedhalmaz valamely részét adatvédelmi megfontolásból külön kívánjuk tárolni.
- Az egyik egyedhalmazban (B) olyan adatokat szeretnénk tárolni, amely a fő egyedhalmazban (A) csak bizonyos elemekre érvényes.

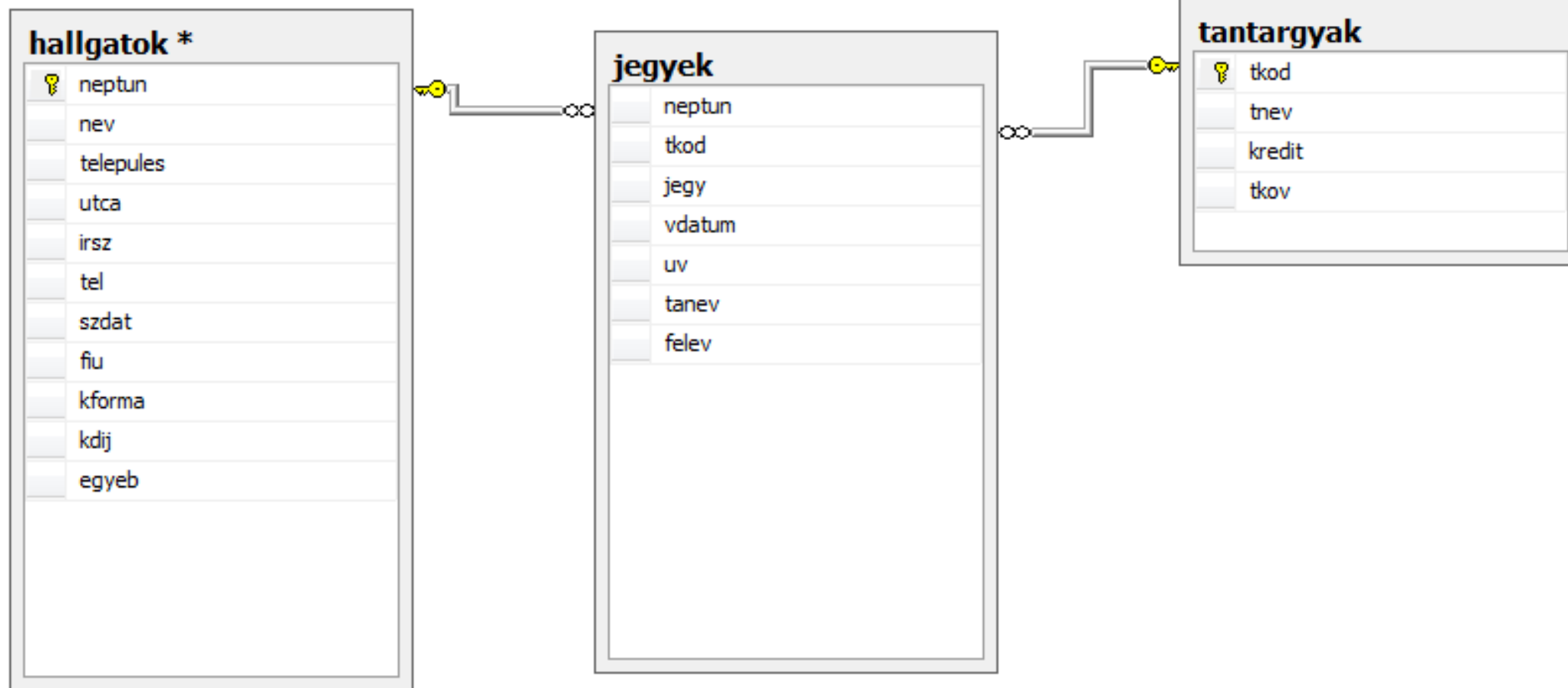
- **Egy-a-többhöz kapcsolat (1 : N) Pl.: Anya és gyermekei**

(A - B) kapcsolat leggyakrabban használatos kapcsolati típus. Az egy-a-többhöz kapcsolatban az A egyedhalmaz valamely eleméhez több elem tartozhat a B egyedhalmazban, de a B egyedhalmaz valamennyi eleméhez csak egy-egy elem tartozhat az A egyedhalmazban.

- **Több-a-többhöz kapcsolat (M : N) Pl.: Dédszülők és dédunokák**

(A - B) kapcsolat esetén az A egyedhalmaz valamely eleméhez több elem is tartozhat a B egyedhalmazban, és a B egyedhalmaz valamely eleméhez is több elem tartozhat az A egyedhalmazban. A több-a-többhöz kapcsolat felbontható két egy-a-többhöz kapcsolatra.

Relációs adatbázis ADATMODELL-je





Objektumok:

adatséma, adattípus, oszloptípus, jelkészlet

► Adatséma:

Az adatbázissémák olyan teljes, vagy részadatbázisok leírásai, amelyekbe tartozó objektumokat csak adott felhasználók, jelszók ismeretében érhetnek el. Tehát az adatsémák **definiálják, hogy adott felhasználók az adatbázis mely részleteit, milyen jogosultságokkal érhetik el.**

Objektumok: adatséma, adattípus, oszloptípus, jelkészlet

Adattípus:

Az SQL öt alapvető adattípust különböztet meg:

| Adattípust | Leírás |
|-----------------------------|--|
| Számszerű | Olyan adatok tárolására szolgál, amelyekkel numerikus műveleteket végezhetünk. |
| Szöveges | Tetszőleges szöveges információ tárolására szolgál. |
| Dátum (idő) jellegű | Dátum és / vagy idő jellegű adatokat tároló adattípus. |
| Bináris vagy logikai | Kétértékű adatok tárolását végzi. |
| | |

Objektumok:

adatséma, **adattípus**, oszloptípus, jelkészlet

| | | | | |
|--------------|-------------------------------|------------------|--------------------------|---|
| bigint | 9.223.372.036.854.775. 808 | - 7 | 9.223.372.036.854.775.80 | 8 |
| int | -2.147.483.648 | | 2.147.483.647 | 4 |
| smallint | -32.768 | | 32.767 | 2 |
| tinyint | 0 | | 255 | 1 |
| bit | 0 | 1, NULL is lehet | | minden megkezdett 8 bit típusú oszlop 1 byte |
| decimal(p,s) | -10 ³⁸ +1 | | 10 ³⁸ -1 | p-től függ |
| numeric(p,s) | -10 ³⁸ +1 | | 10 ³⁸ -1 | min. 5, max. 17 byte |
| money | 922.337.203.685.477,58 08 | - 7 | +922.337.203.685.477,580 | 8 |
| smallmoney | -214.748.3648 | | +214.748.3647 | 4 |

| Típus | Leírás |
|--------------|--|
| char | Fix hosszú nem Unicode kar., maximum 8.000 kar. |
| varchar | Változó hosszú nem Unicode kar., maximum 8.000 kar. |
| varchar(max) | Változó hosszú nem Unicode kar., maximum 2 ³¹ kar (csak SQL Server 2005). |
| text | Változó hosszú nem Unicode kar., maximum 2.147.483.647 kar. |



Objektumok: adatséma, adattípus, oszloptípus , jelkészlet

► Oszloptípus (DOMAIN):

Az alap adattípusokra építve definiálhatunk saját adattípust, melyet az adattábla mezőinek adattípus definíciójában szintén használhatunk. Ha módosítunk az oszloptípus definíción, akkor az automatikusan tükröződik minden olyan tábla szerkezetében, ahol felhasználtuk mező adattípus definiálására.

Az oszloptípus definiálásakor az adattípus, méret jellemzők mellett **megadhatunk alapértelmezett értéket, ellenőrzési szabályt**. Az alapértelmezett érték automatikusan bekerül a tábla újonnan felvitt sorába. Az adatmódosítást vizsgálja az ellenőrzési szabály, a tiltott módosítást nem engedi bevinni a mezőbe

Objektumok:

adatséma, adattípus, oszloptípus, **jelkészlet**

Jelkészlet

A rendszer által kezelt jelek halmaza,
operációsrendszer függő.

A leggyakrabban használatos
jelkészlet az ASCII.

| Binary | Oct | Dec | Hex | Glyph | Binary | Oct | Dec | Hex | Glyph | Binary | Oct | Dec | Hex | Glyph |
|----------|-----|-----|-----|-------|----------|-----|-----|-----|-------|----------|-----|-----|-----|-------|
| 010 0000 | 040 | 32 | 20 | sp | 100 0000 | 100 | 64 | 40 | @ | 110 0000 | 140 | 96 | 60 | ` |
| 010 0001 | 041 | 33 | 21 | ! | 100 0001 | 101 | 65 | 41 | A | 110 0001 | 141 | 97 | 61 | a |
| 010 0010 | 042 | 34 | 22 | " | 100 0010 | 102 | 66 | 42 | B | 110 0010 | 142 | 98 | 62 | b |
| 010 0011 | 043 | 35 | 23 | # | 100 0011 | 103 | 67 | 43 | C | 110 0011 | 143 | 99 | 63 | c |
| 010 0100 | 044 | 36 | 24 | \$ | 100 0100 | 104 | 68 | 44 | D | 110 0100 | 144 | 100 | 64 | d |
| 010 0101 | 045 | 37 | 25 | % | 100 0101 | 105 | 69 | 45 | E | 110 0101 | 145 | 101 | 65 | e |
| 010 0110 | 046 | 38 | 26 | & | 100 0110 | 106 | 70 | 46 | F | 110 0110 | 146 | 102 | 66 | f |
| 010 0111 | 047 | 39 | 27 | ' | 100 0111 | 107 | 71 | 47 | G | 110 0111 | 147 | 103 | 67 | g |
| 010 1000 | 050 | 40 | 28 | (| 100 1000 | 110 | 72 | 48 | H | 110 1000 | 150 | 104 | 68 | h |
| 010 1001 | 051 | 41 | 29 |) | 100 1001 | 111 | 73 | 49 | I | 110 1001 | 151 | 105 | 69 | i |
| 010 1010 | 052 | 42 | 2A | * | 100 1010 | 112 | 74 | 4A | J | 110 1010 | 152 | 106 | 6A | j |
| 010 1011 | 053 | 43 | 2B | + | 100 1011 | 113 | 75 | 4B | K | 110 1011 | 153 | 107 | 6B | k |
| 010 1100 | 054 | 44 | 2C | , | 100 1100 | 114 | 76 | 4C | L | 110 1100 | 154 | 108 | 6C | l |
| 010 1101 | 055 | 45 | 2D | - | 100 1101 | 115 | 77 | 4D | M | 110 1101 | 155 | 109 | 6D | m |
| 010 1110 | 056 | 46 | 2E | . | 100 1110 | 116 | 78 | 4E | N | 110 1110 | 156 | 110 | 6E | n |
| 010 1111 | 057 | 47 | 2F | / | 100 1111 | 117 | 79 | 4F | O | 110 1111 | 157 | 111 | 6F | o |
| 011 0000 | 060 | 48 | 30 | 0 | 101 0000 | 120 | 80 | 50 | P | 111 0000 | 160 | 112 | 70 | p |
| 011 0001 | 061 | 49 | 31 | 1 | 101 0001 | 121 | 81 | 51 | Q | 111 0001 | 161 | 113 | 71 | q |
| 011 0010 | 062 | 50 | 32 | 2 | 101 0010 | 122 | 82 | 52 | R | 111 0010 | 162 | 114 | 72 | r |
| 011 0011 | 063 | 51 | 33 | 3 | 101 0011 | 123 | 83 | 53 | S | 111 0011 | 163 | 115 | 73 | s |
| 011 0100 | 064 | 52 | 34 | 4 | 101 0100 | 124 | 84 | 54 | T | 111 0100 | 164 | 116 | 74 | t |
| 011 0101 | 065 | 53 | 35 | 5 | 101 0101 | 125 | 85 | 55 | U | 111 0101 | 165 | 117 | 75 | u |
| 011 0110 | 066 | 54 | 36 | 6 | 101 0110 | 126 | 86 | 56 | V | 111 0110 | 166 | 118 | 76 | v |
| 011 0111 | 067 | 55 | 37 | 7 | 101 0111 | 127 | 87 | 57 | W | 111 0111 | 167 | 119 | 77 | w |
| 011 1000 | 070 | 56 | 38 | 8 | 101 1000 | 130 | 88 | 58 | X | 111 1000 | 170 | 120 | 78 | x |
| 011 1001 | 071 | 57 | 39 | 9 | 101 1001 | 131 | 89 | 59 | Y | 111 1001 | 171 | 121 | 79 | y |
| 011 1010 | 072 | 58 | 3A | : | 101 1010 | 132 | 90 | 5A | Z | 111 1010 | 172 | 122 | 7A | z |
| 011 1011 | 073 | 59 | 3B | ; | 101 1011 | 133 | 91 | 5B | [| 111 1011 | 173 | 123 | 7B | { |
| 011 1100 | 074 | 60 | 3C | < | 101 1100 | 134 | 92 | 5C | \ | 111 1100 | 174 | 124 | 7C | |
| 011 1101 | 075 | 61 | 3D | = | 101 1101 | 135 | 93 | 5D |] | 111 1101 | 175 | 125 | 7D | } |
| 011 1110 | 076 | 62 | 3E | > | 101 1110 | 136 | 94 | 5E | ^ | 111 1110 | 176 | 126 | 7E | ~ |
| 011 1111 | 077 | 63 | 3F | ? | 101 1111 | 137 | 95 | 5F | _ | | | | | |

További CONSTRAINT osztályok...

- NOT NULL: Kötelező értéket megadni az oszlopban.
- CHECK: Kikényszeríti a tartományintegritást az oszlopban megadható értékek korlátozásával. Meghatároz egy logikai kifejezést, mely kiértékelődik, amikor megadunk egy oszlopértéket. Ha a kiértékelés hamis eredményt hoz, akkor nem engedi felvinni a rendszer az új értéket az oszlopba. Egy oszlophoz több CHECK constraint is rendelhető.
- UNIQUE: Biztosítja, hogy ha nem NULL egy adott oszlopérték, akkor egyedi.
- PRIMARY KEY: megadhat egy vagy több oszlopot, melyek értékei egyedien azonosítanak egy táblabeli sort. Nem engedi meg a NULL érték használatát. Táblánként csak egy elsődleges kulcs lehet. (de az akár lehet összetett is)
- FOREIGN KEY: Kapcsolatot biztosít két tábla között. Az egyik tábla idegen kulcsa a másik tábla jelölt kulcsára mutat. Megakadályozza olyan kulcsérték bevitelét, amely nem fordul elő a jelölt kulcs értékei között. ON DELETE záradékkal megadhatjuk, hogy mi történjen az idegenkulcsok soraival, ha a jelölt kulcs sora törlődik. ON UPDATE szintén.

Objektumok:

adattábla, index tábla, nézettábla

Adattábla (Table):

Az adatbázis alapvető objektuma. Táblázatos formában tárolja az egyedtípus előfordulásokat. A táblázat oszlopai (mezői) az egyedhez definiált tulajdonságtípusok.

Oszloponként legalább meg kell adni a nevet, adattípust, méretet. Ezen kívül az SQL2 további mezőjellemzőket (indexelt, NULL érték használat, alapértelmezett érték, adatérték-szabály (CONSTRAINT), stb.) is támogat.

Hallgatók tábla

| | |
|-----------|---------------------------|
| neptun | [nchar](6) NOT NULL, |
| nev | [nvarchar](50) NOT NULL, |
| telepules | [nvarchar](50) NOT NULL, |
| utca | [nvarchar](50) NOT NULL, |
| irsz | [nchar](10) NULL, |
| tel | [nvarchar](50) NULL, |
| szdat | [smalldatetime] NOT NULL, |
| fiu | [bit] NULL, |
| kforma | [nchar](10) NULL, |
| kdijs | [money] NULL, |
| egyeb | [nvarchar](max) NULL, |

Objektumok: adattábla, **index tábla**, nézettábla

Indextábla (INDEX):

Adattáblára alapozva definiálhatunk indexeket. Az indextábla az indexkulcsokat (kimásolva az adattáblából) és a kulcshoz tartozó sor (rekord) fizikai címét tartalmazza rendezett formában, így a kulcsszerinti keresés nagyon gyors.

A rendezettség iránya lehet növekvő, csökkenő. Az indexkulcs lehet összetett, több mező együtt adja az indexkulcsot. Egyedi indexek esetén egy tetszőleges kulcsérték csak egyszer fordul elő. Adattáblánként egy Primary Key (Elsődleges kulcs, egyedi index) definiálható, amely a táblák közötti kapcsolatok létrehozásánál játszik fontos szerepet. A primary key index definiálása az adattábla szerkezetének megadásakor, vagy módosításakor történik. A rendszer automatikusan kezeli az indextáblákat. Ha egy rendezettségre igény jelentkezik, akkor megnézi, hogy van-e olyan indexe. Ha igen, akkor használja, ha nem akkor elvégzi a szükséges rendezést. Az adattáblával együtt az indextáblák automatikusan karbantartódnak.



Objektumok: adattábla, index tábla, **nézettábla**

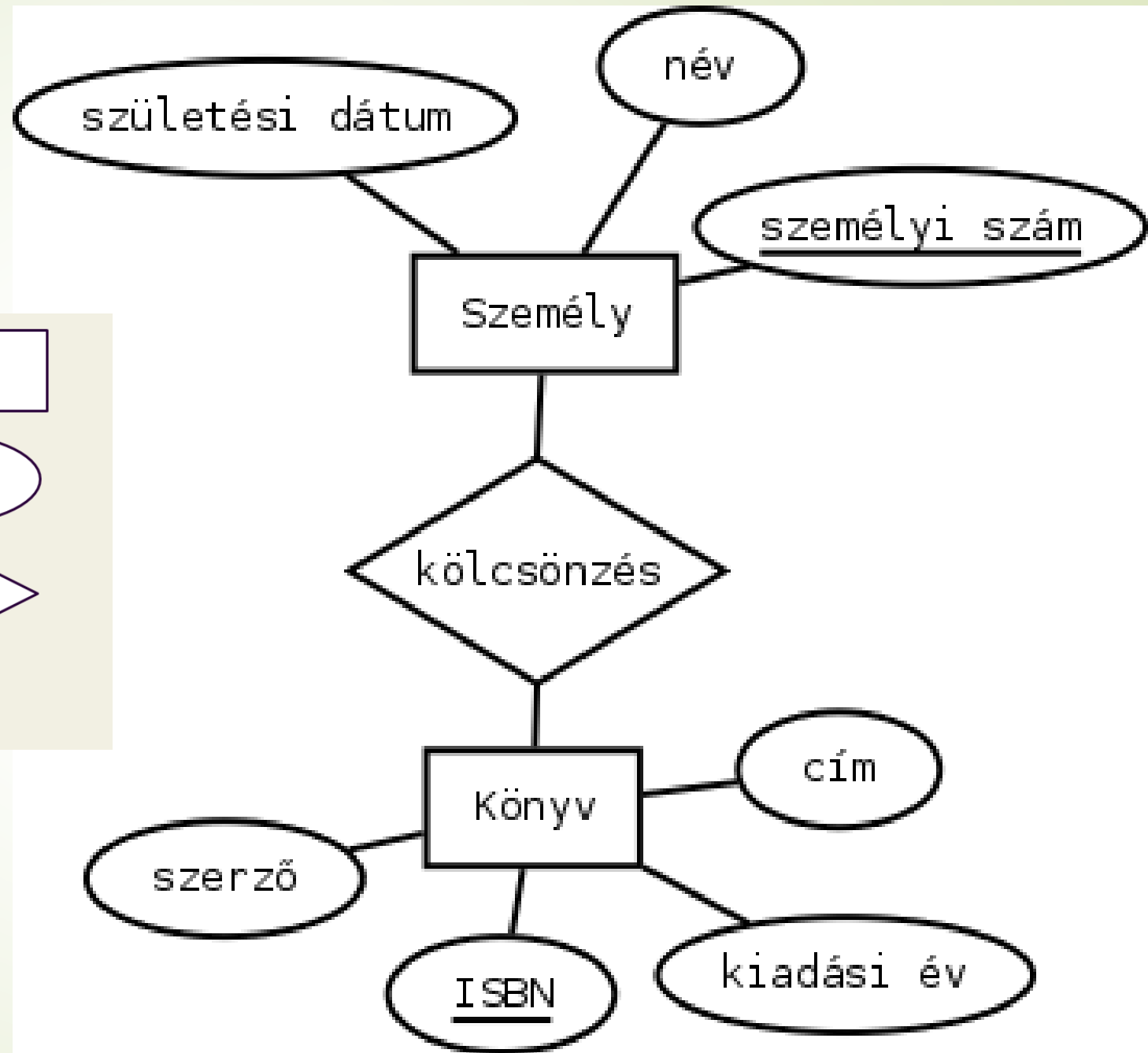
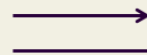
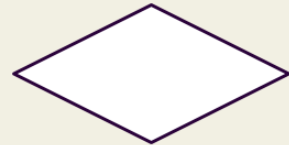
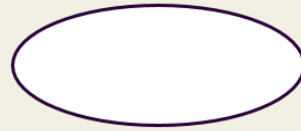
Nézettábla (VIEW):

Adattáblákra, korábban definiált nézettáblákra alapozva egy lekérdezést (SELECT) definiálhatunk. Csak a definíció kerül tárolásra. Amikor a nézettáblára hivatkozunk, akkor a rendszer a definíció alapján előállítja a lekérdezést. Ezután a lekérdezés eredménytáblában történő módosítás is. Lehetséges a nézettábla alapján az adattáblában történő módosítás is. Az adatséma jellemző objektumai a nézetek.

EK modell

Adatmodell sematikus ábrája

- Egyedhalmaz (táblák)
- Attribútum (sorok)
- Kapcsolat (közös kulcs)
- Összeköttetés



CODD szabályai

- ▶ Edgar F. Codd 12 szabály néven elhíresült 13 megállapítása a relációs adatbázis-kezelő rendszerek legfontosabb ismérveit rögzítik. (1985-ből)
- ▶ **A 0. szabály**
Ahhoz, hogy egy rendszer **relációs**nak, **adatbázis**nak, és **kezelő rendszer**nek legyen nevezhető, elfogadható – a rendszernek a relációkkal foglalkozó adottságait kizárólag az adatbázis kezelésére kell használnia.
- ▶ **1. Az egységes megjelenésű információ szabálya**
Ez a szabály egyszerűen csak annyit ír elő, hogy az adatbázisban szereplő összes információt egy, és csak egy megadott formában lehet ábrázolni, nevezetesen a sorokba szedett táblázatokon belül egy-egy oszlop pozícióban. (vízszintes és függőleges koordináta-rendszerben)

CODD szabályai

➤ 2. Garantált lokalizálhatóság szabálya

Elsődleges kulcs alapkövetelménye: Azt mondja ki, hogy az adatbázisban minden egyes skaláris értékre logikailag úgy kell hivatkozni, hogy megadjuk az azt tartalmazó táblázat és az oszlop nevét, valamint a megfelelő sor (azt tartalmazó sor) elsődleges kulcsának az értékét.

➤ 3. A NULL értékek egységes kezelése

Az adatbázis-kezelő rendszernek olyan módszerrel kell támogatnia a „hiányzó és nem felhasználható információt” amely egységes, és eltér az összes „rendes” érték kezelésétől (például numerikus értékek esetében, „nullától vagy más számtól különböző”-ként), továbbá független az adattípustól. Ebbe az is beletartozik, hogy ezeknek a reprezentációknak a kezelését a szoftvernek módszeresen kell végeznie.

CODD szabályai

- ▶ **4. A relációs modell alapján aktív online katalógust kell üzemben tartani**
A rendszernek támogatnia kell egy online, beépített katalógust, amely a szokásos lekérdező nyelvet használó feljogosított felhasználók előtt nyitva áll.
- ▶ **5. A teljes körű „adatnyelv” szabálya**
A rendszernek legalább egy olyan relációs nyelvet kell támogatnia, amelynek
 - ▶ (a) lineáris a szintaxisa,
 - ▶ (b) interaktívan és alkalmazási programokon belül is lehet használni, továbbá
 - ▶ (c) támogatja az adat definiáló műveleteket (beleértve az adatok megjelenítési képeinek meghatározására szolgálókat), az adatmódosító (manipulációs) műveleteket (frissítés és visszakeresés is), biztonsági és jósági (integritási) korlátokat, valamint a tranzakció kezelési műveleteket (begin, commit, és rollback: elkezdés, jóváhagyás és visszagörgetés).

CODD szabályai

➤ 6. A nézetek frissítésének szabálya

A rendszernek képesnek kell lennie az adatok elméletileg frissíthető minden nézetének frissítésére.

➤ 7. Magas szintű beszúrás, frissítés és törlés

A rendszernek támogatnia kell az INSERT, UPDATE, és DELETE (új adat, módosítás, törlés) operátorok halmaz szintű, egyidejű működését.

➤ 8. Fizikai szintű adatfüggetlenség

A fizikai adatfüggetlenség akkor áll fenn, ha az alkalmazások (programok) és a felhasználók adatelérési módja független az adatok tényleges (fizikai) tárolási és elérési módjától.

CODD szabályai

➤ 9. Logikai szintű adatfüggetlenség

Logikai adatfüggetlenség akkor áll fenn, ha az adatbázis logikai szerkezetének megváltoztatása nem igényli az adatbázist használó alkalmazások (programok) megváltoztatását.

➤ 10. Jóság (integritás) függetlenség

Az adatok jóságának (érvényességének) korlátait az adatfeldolgozási programoktól függetlenül kell tudni meghatározni, és azokat katalógusban kell nyilvántartani. Legyen lehetséges a szóban forgó korlátokat – úgy és amikor szükséges – megváltoztatni, anélkül hogy a meglévő alkalmazásokat szükségtelen módon zavarnánk.

CODD szabályai

➤ 11. Elosztástól való függetlenség

A meglévő alkalmazások működése zavartalan kell, hogy maradjon

- (a) amikor sor kerül az adatbázis kezelő szoftver elosztott változtatásnak bevezetésére a rendszerben
- (b) amikor a meglévő elosztott adatokat a rendszer újra szétosztja

➤ 12. Megkerülhetetlenség szabálya

Ha a rendszernek van egy alacsony szintű (egyszerre egy rekordot érintő) interfésze, akkor azt az interfészt ne lehessen a rendszer megkerülésére használni, például a relációs biztonsági vagy jósági (integritás védelmi) korlátok megsértésével.

Relációs adatbázis kezelő fő részei

► Lekérdezés feldolgozó

A magas szintű programnyelven megfogalmazott (pl.: SQL nyelv) lekérdezéseket, adatbázis-műveleteket egyszerű utasítások sorozatává alakítja.

► Tárkezelő

Az adatbázis-kezelők általában közvetlenül felügyelik az adatok lemezen való tárolását.

A tárkezelő két részből áll:

- **Fájlkezelő:** Nyilvántartja a fájlok lemezen való elhelyezkedését és beolvassa egy állomány blokkjait a puffer kezelő kérésére. Az állományok lemezblokkokból épülnek fel, méretük általában 4, 8, vagy 16kb.
- **Puffer kezelő:** A memóriát kezeli. A lemezről beolvasott blokkokat egy memóriaterületen (puffer) tárolja. A különböző műveletek innen kezelik az adatokat. Ha betelt a puffer terület, akkor a „nem használt” blokkokat visszaírja a lemezre, így a felszabaduló memóriaterületre új blokkokat olvas. „Kérésre” a blokkokat szintén lemezre írja.

Relációs adatbázis kezelő fő részei

Tranzakció-kezelő

Tranzakció: Olyan műveletek csoportja, amelyeket egymás után egy egységként kell végrehajtani. Vagy az összes műveletet végrehajtjuk, vagy egyet sem.

A tranzakció-kezelő feladata a tranzakciók helyes lefutásának biztosítása.

Alapvető elvárások egy tranzakció-kezelővel szemben:

- **Atomosság:** A tranzakció vagy teljes egészében hajtódjon végre, vagy egyáltalán ne. Pl.: pénzautomata...
- **Konzisztencia/Következetesség:** Az adatok megfelelnek bizonyos elvárásoknak. Pl.: Egy repülőgép-helyfoglalási adatbázisban feltétel, hogy egyetlen ülőhelyet se rendeljünk hozzá két különböző utashoz. A feltételt megsérthetjük egy rövid időre a tranzakció alatt (Pl.: Amíg az utasokat áthelyezzük az ülőhelyek között), a tranzakció-kezelő biztosítja, hogy a tranzakciók befejeződése után az adatbázis ismét következetes állapotba kerüljön.
- **Izoláció/Elkülönítés:** **Az egyidejűleg futó tranzakciók egymásra hatását el kell különíteni egymástól**, mintha egymás után futnának. Ha két ügyintéző ugyanarra a járatra ad el jegyet és a járaton már csak egy hely van, akkor az egyik kérést teljesíteni kell, a másikat elutasítani.
- **Tartósság:** Ha egy tranzakció befejezte a munkáját akkor annak eredménye nem veszt el rendszerhiba esetén, akkor sem, ha a rendszer közvetlenül a tranzakció befejezése után hibásodik meg.

Normálformák: 1NF

(első normálforma)

Cél: **Teljes funkcionális függőségek legyenek.**
Kialakítása

- Egyedeinket -> egyedtípusokba rendezzük
- Tulajdonságaikat → tulajdonságtípusokba rendezzük.

Feltételek:

- Minden rekord különbözik
- Rekordonként megegyezik a mezők száma, és sorrendje
- Nincsenek többértékű mezők.

| Személyek | | | | | | | |
|------------|---------------|------|-------------|--------------|--|--------------------|---|
| SzemlgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 +36/70/788 9130 +36/20/315 5551 | Igen Nem Nem | 08:00-13:00 09:30-18:00 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3000 | Hatvan | Cukor út 12. | +36/36/321 321 +36/30/368 5552 | Igen Nem | 08:00-13:00 08:00-20:00 |

Normálformák: 1NF

(első normálforma)

A jó megoldást a következő változat nyújtja. Ebben a táblában a több telefonnal rendelkező személyeket többször is eltároltuk, annak megfelelően, hogy hány telefonnal rendelkeznek. Így elértük, hogy ne legyenek többértékű mezők.

| Személyek | | | | | | | |
|------------|---------------|------|-------------|--------------|-----------------|-----------|-------------|
| SzemlgSzam | Nev | IRSZ | Varos | Utca | Szam | Vezetekes | Mikor |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. | +36/20/324 7889 | Nem | 08:00-16:00 |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gornai u. 1. | +36/30/678 6634 | Nem | 00:00-24:00 |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 | +36/1/320 5122 | Igen | 18:00-22:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/36/321 321 | Igen | 08:00-13:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/70/788 9130 | Nem | 09:30-18:00 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. | +36/20/315 5551 | Nem | 08:00-20:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/36/321 321 | Igen | 08:00-13:00 |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. | +36/30/368 5552 | Nem | 08:00-20:00 |
| 060811 DI | Örök Virág | 3100 | Salgótarján | Főút 32. | +36/32/932928 | Igen | 19:00-20:30 |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. | +36/36/555 325 | Igen | 18:00-20:00 |

Normálformák: 2NF

(második normálforma)

A 2NF előfeltétele, hogy

- adatbázisunk minden táblája legalább 1NF-ben legyen!
- a reláció minden nem elsődleges attribútuma teljes funkcionális függőségben van az összes reláció kulccsal.

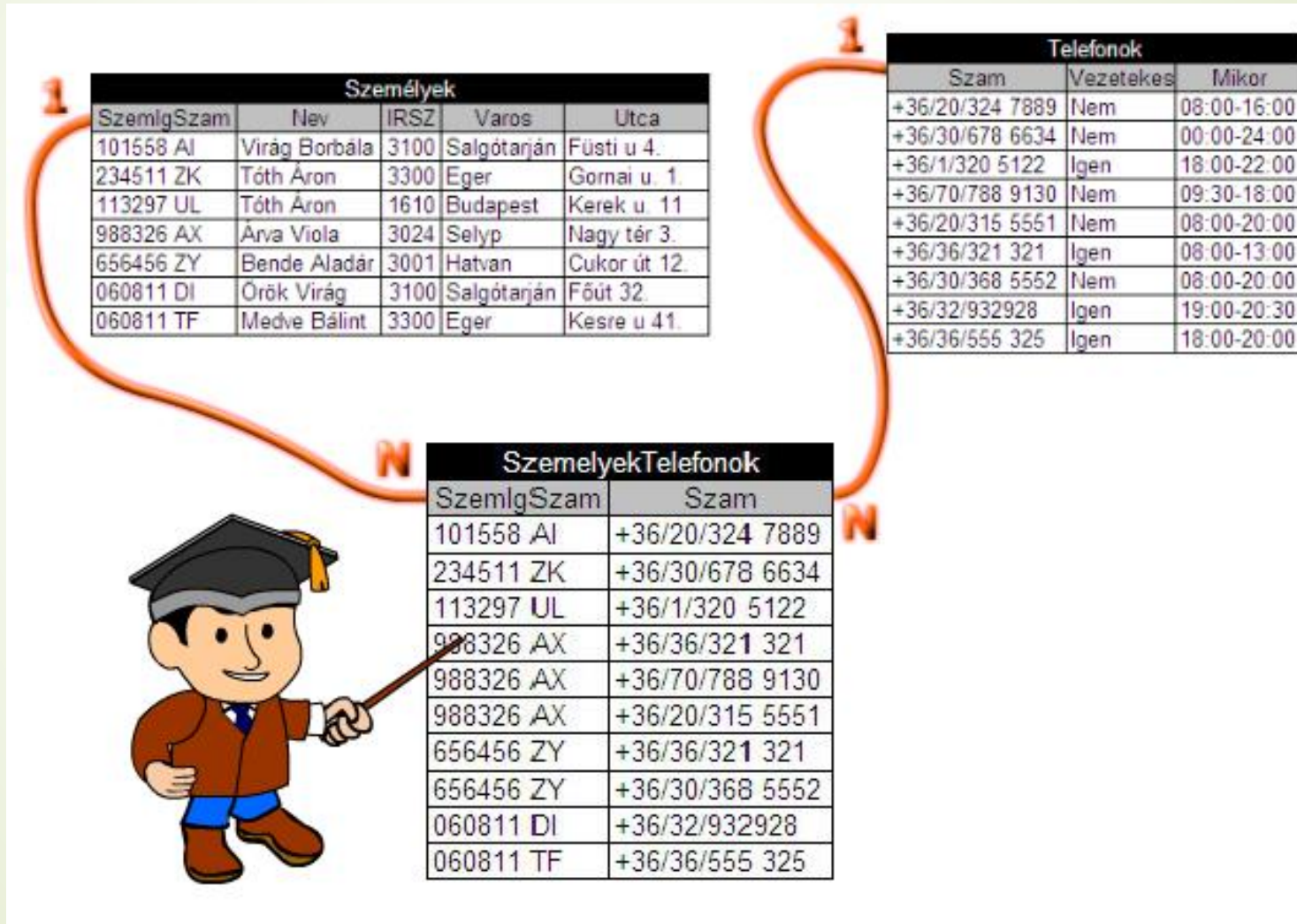
Tehát a táblákban **megszüntetjük az esetleges részleges funkcionális függéseket.**

| Személyek | | | | |
|------------|---------------|------|-------------|--------------|
| SzemlgSzam | Nev | IRSZ | Varos | Utca |
| 101558 AI | Virág Borbála | 3100 | Salgótarján | Füsti u 4. |
| 234511 ZK | Tóth Áron | 3300 | Eger | Gomai u. 1. |
| 113297 UL | Tóth Áron | 1610 | Budapest | Kerek u. 11 |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. |
| 988326 AX | Árva Viola | 3024 | Selyp | Nagy tér 3. |
| 656456 ZY | Bende Aladár | 3000 | Hatvan | Cukor út 12. |
| 656456 ZY | Bende Aladár | 3001 | Hatvan | Cukor út 12. |
| 060811 DI | Órök Virág | 3100 | Salgótarján | Főút 32. |
| 060811 TF | Medve Bálint | 3300 | Eger | Kesre u 41. |

| Telefonok | | |
|-----------------|-----------|-------------|
| Szam | Vezetekes | Mikor |
| +36/20/324 7889 | Nem | 08:00-16:00 |
| +36/30/678 6634 | Nem | 00:00-24:00 |
| +36/1/320 5122 | Igen | 18:00-22:00 |
| +36/36/321 321 | Igen | 08:00-13:00 |
| +36/70/788 9130 | Nem | 09:30-18:00 |
| +36/20/315 5551 | Nem | 08:00-20:00 |
| +36/36/321 321 | Igen | 08:00-13:00 |
| +36/30/368 5552 | Nem | 08:00-20:00 |
| +36/32/932928 | Igen | 19:00-20:30 |
| +36/36/555 325 | Igen | 18:00-20:00 |

Normálformák: 2NF

(második normálforma)



Normálformák: 3NF

(harmadik normálforma)

A 3NF előfeltétele, hogy :

- ▶ adatbázisunk minden táblája legalább 2NF-ben legyen!
- ▶ A reláció nem tartalmaz funkcionális függőséget a nem elsődleges attribútumok között.

Tehát : a táblákban megszüntetjük a **tranzitív függéseket**.

| Szemelyek | | | |
|------------|---------------|--------------|------|
| SzemlgSzam | Nev | Utca | IRSZ |
| 101558 AI | Virág Borbála | Füsti u 4. | 3100 |
| 234511 ZK | Tóth Aron | Gornai u. 1. | 3300 |
| 113297 UL | Tóth Aron | Kerek u. 11 | 1610 |
| 988326 AX | Árva Viola | Nagy tér 3. | 3024 |
| 656456 ZY | Bende Aladár | Cukor út 12. | 3001 |
| 060811 DI | Örök Virág | Főút 32. | 3100 |
| 060811 TF | Medve Bálint | Kesre u 41. | 3300 |

| Varosok | |
|---------|-------------|
| IRSZ | Varos |
| 3100 | Salgótarján |
| 3300 | Eger |
| 1610 | Budapest |
| 3024 | Selyp |
| 3001 | Hatvan |

| SzemelyekTelefonok | |
|--------------------|-----------------|
| SzemlgSzam | Szam |
| 101558 AI | +36/20/324 7889 |
| 234511 ZK | +36/30/678 6634 |
| 113297 UL | +36/1/320 5122 |
| 988326 AX | +36/36/321 321 |
| 988326 AX | +36/70/788 9130 |
| 988326 AX | +36/20/315 5551 |
| 656456 ZY | +36/36/321 321 |
| 656456 ZY | +36/30/368 5552 |
| 060811 DI | +36/32/932928 |
| 060811 TF | +36/36/555 325 |

| Telefonok | | |
|-----------------|-----------|-------------|
| Szam | Vezetekes | Mikor |
| +36/20/324 7889 | Nem | 08:00-16:00 |
| +36/30/678 6634 | Nem | 00:00-24:00 |
| +36/1/320 5122 | Igen | 18:00-22:00 |
| +36/70/788 9130 | Nem | 09:30-18:00 |
| +36/20/315 5551 | Nem | 08:00-20:00 |
| +36/36/321 321 | Igen | 08:00-13:00 |
| +36/30/368 5552 | Nem | 08:00-20:00 |
| +36/32/932928 | Igen | 19:00-20:30 |
| +36/36/555 325 | Igen | 18:00-20:00 |

Adatok kezelése az adatbázisban = SQL

Az adatok gyors és egyszerű kereshetősége létfontosságú napjainkban.

A relációs adatbázisok kezeléséhez teremtett megfelelő keretrendszer az egyik legelterjedtebb programozási nyelv, a Structured Query Language.

A relációs adatbázis-kezelő rendszer

- A relációsadatbázis-kezelő rendszer (angol rövidítéséből: RDBMS) egy olyan adatbázis-kezelő rendszer, amelynek logikai adatbázisát szoftverkomponensei kizárólag a relációs adatmodellek elvén épülnek fel, illetve kérdezhetők le.
- A relációsadatbázis-kezelő rendszerek szabványos adat hozzáférési és adatdefiníciós nyelve az **SQL**.
- **FŐGONOSZ***: IBM



SQL

- ▶ Az SQL, azaz Structured Query Language (strukturált lekérdezőnyelv) relációsadatbázis-kezelők lekérdezési nyelve.
- ▶ Angol nyelvterületen 'eszkjuel' a kiejtése. A hagyományokhoz való hűség jegyében sokan 'szikvel'-nek ejtik, ugyanis korábban Structured English Query Language (SEQUEL) volt az elnevezés, és ezt rövidítették le.



SQL nyelv

Az SQL nyelvi elemeket 4 részre lehet bontani:

- adatdefiníciós (Data Definition Language, DDL),
- adatkezelési (Data Manipulation Language, DML),
- lekérdező (QUERY Language, QL)) és
- adatvezérlő (Data Control Language, DCL)



DDL (Data Definition Language)

Adatbázis és táblái létrehozása.
Relációk, indexek szerkezetének kialakítása,
módosítása.

CREATE – létrehozás

ALTER – módosítás

DROP - törlés



DML (Data Manipulation Language)

- Relációk adatainak felvitele, módosítása, vagy törlése

INSERT – adatok beszúrása

UPDATE – adatok módosítása, frissítése

DELETE – adatok törlése relációból



QL (QUERY Language)

- ▶ Relációk adatainak lekérdezése

SELECT (DISTINCT/TOP) mezőnevek, függvények, összefűzés

FROM tábla nevek

WHERE feltétel(ek)

GROUP BY csoportosítási mezők

HAVING csoportosítási feltételek, utószűrés

ORDER BY rendezési mezők



DCL (Data Control Language)

Felhasználók jogosultságait lehet vele paraméterezni, azaz hogy egy adatbázis esetében milyen műveleteket használhat egy felhasználó

GRANT – jogok adása

REVOKE – jogok visszaállítása

Bevezetés az SQL-be



<http://sqlfiddle.com/>

Bevezetés az SQL-be

SÉMA (SCRIPT)

```
create table termekek(  
id int primary key identity,  
megnevezes nvarchar(80),  
mennyiseg int,  
reszleg varchar (3)  
)
```

```
insert termekek (megnevezes, mennyiseg, reszleg) values ('Borsodi', 3, 'ALK')
```

```
insert termekek (megnevezes, mennyiseg, reszleg) values ('Amstel', 8, 'AUT')
```

```
insert termekek (megnevezes, mennyiseg, reszleg) values ('Coronita', 6, 'ALK')
```

```
insert termekek (megnevezes, mennyiseg, reszleg) values ('Kőbányai', 6, 'AUT')
```

```
insert termekek (megnevezes, mennyiseg, reszleg) values ('Kozel', 24, 'ALK')
```

```
insert termekek (megnevezes, mennyiseg, reszleg) values ('Stella Artois', 20, 'AUT')
```

- SELECT
- FROM
- WHERE
- GROUP BY
- HAVING
- ORDER BY

Google

sql injection licence plate



HI, THIS IS
YOUR SON'S SCHOOL.
WE'RE HAVING SOME
COMPUTER TROUBLE.



OH, DEAR - DID HE
BREAK SOMETHING?

IN A WAY-)



DID YOU REALLY
NAME YOUR SON
Robert'); DROP
TABLE Students;-- ?



OH, YES. LITTLE
BOBBY TABLES,
WE CALL HIM.

WELL, WE'VE LOST THIS
YEAR'S STUDENT RECORDS.
I HOPE YOU'RE HAPPY.



AND I HOPE
YOU'VE LEARNED
TO SANITIZE YOUR
DATABASE INPUTS.



Köszönöm a figyelmet!

