

Adatbázis-kezelés

Takács Gábor

mérnökinformatikus, okl. mérnök tanár
Informatika Tanszék, mesteroktató

takacsg@sze.hu

<http://takacsg.hu/>



Gyakorlatok átjárhatósága

- IGEN, de jelenléti ívre figyelni.
- ZH csak abban a kurzusban írható, ahova a hallgató fel van iratkozva neptunban.

Letölthető anyagok: takacsg.hu

Nappali tantárgyak (SZE – GYŐR):

<p>Adatbázis-kezelés 1. (GKNB_INTM123)</p>		<p><u>Téma regisztrációs űrlap</u></p> <p>2024.09.13 – 12:00</p> <p>-----</p> <p>Féléves leadás:</p> <p><u>Fájlfeltöltő LINK</u></p> <p>plusz pontért:</p> <p>2024.10.20. – 12:00</p> <p>végleges határidő:</p> <p>2024.10.31 – 12:00</p>
--	---	---

Féléves feladat kapcsán elvégzendő feladatok

FÉLVES FELADAT KÉSZÍTÉSE: 20-30 oldal terjedelemben

Cél: Adatáramlást biztosítani tudó adatbázis tervezése, megvalósítása

- Borító, és tartalomjegyzék készítése
- Legalább 6 táblás adatbázis támogatással megvalósítandó program szöveges ismertetése, bemutatása
- Adathalmaz összegyűjtése: input, output és számított adatok meghatározása
- Relációs sémák elkészítése
- Adatmodell táblázatos bemutatása
(tulajdonság neve, adattípusa, NULL értéke, egy mondatos leírása)
- Adatmodell bemutatása
- Egyed-Kapcsolat diagram elkészítése
- Megszorítási szabályok szöveges bemutatása, megfogalmazása
- 2-3 GUI elkészítése az elképzelt alkalmazáshoz, programhoz
- Mellékletbe: MSSQL-ben elkészített adatbázis export scriptjének bemásolása

Féléves feladat sablon

A féléves feladat készítésének követelményei:

Valós feladat esetén a féléves feladat titkos, bizalmas információt nem tartalmazhat

- 1 oldal: Borítólap
- 2. oldal: Tartalomjegyzék
- 3. oldaltól jön maga a féléves:
 - Formázás: Arial betűtípus, 12-es betűméret, sorkizárt, 1.0-es sorközzel
 - Margók: körbe 2,5 cm.

Az első oldalon a ZÖLD részek átírandóak, és átszínezendők fekete színre. Az második oldal után következő kék tartalom átolvasás után törlendő.

Az elkészült adatbázisba táblánként elég 5-6 rekordot feltölteni. A feltöltött adatok is szerepeljenek a mellékletbe illesztendő, generált scriptben.

Féléves feladat címe

Féléves feladat Adatbázis-kezelés 1. tárgyból

HALLGATÓ NEVE, NEPTUN KÓDJA
Szak neve, szintje, tagozata.
Pl.: Mérnökinformaticus, BSc, nappali

készítés dátuma: ÉÉÉÉ.HH.NN formátumban

Féléves feladat sablon

1. Bevezetés

1.1. Feladat ismertetés

Innen jön a maga a féléves feladat és az oldalszámozás is innen kezdődjön.

Kérjük megfogalmazni és részletesen ismertetni, hogy az adatbázis milyen feladatot, folyamatot fog támogatni. Ezen folyamatokban milyen dokumentumok keletkezhetnek. Leíró jelleggel, szövegesen, mondatokban írja le, hogy az adatbázisban milyen szervezet, nagyjából milyen adatait, és milyen funkciókra szeretnék használni. Pl.: tagnyilvántartás, költségvetés elszámolás, sportolói fejlődés követés, stb.

Ha van, akkor mutassa be röviden a (fiktív) szervezetet, akinek az adatbázis készül, hogy mivel foglalkozik, milyen munkát végez, kitérve munkamódszereire, eredményekre.

1.2. Funkcionális követelmények:

Programot nem kell fejleszteni, de próbáljuk felsorolni az elképzelt program funkcionális követelményeit, melyhez az adatbázis készülni fog. Elég felsorolás jelleggel.

Pl:

- Jogosultság kezelés
- Számla generálása nyomtatáskész állapotra
- Felhasználók adatainak kezelése,
- stb.

1.3. Adathalmaz

Ide kérjük vesszővel elválasztani ömlesztve összegyűjteni azokat az tulajdonság típusokat (adatokat), amiket az adatbázis vélhetően kezelni fog. Input, output adatok, és számított értékek (Pl. elmúlt 1 év átlag fizetése a dolgozóknak)

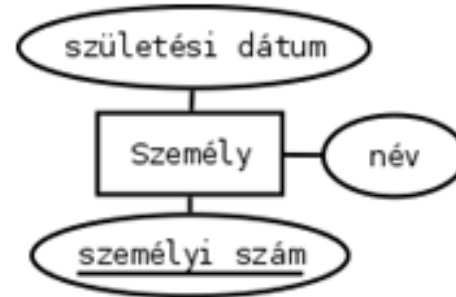
Féléves feladat sablon

2. Egyed-kapcsolat diagram

Az adatbázis logikai modelljének elkészítéséhez az alábbi szempontokat kell figyelembe venni: Miről szeretnénk eltárolni adatokat? Milyen adatokat szeretnénk tárolni? (erre a kettő kérdésre segít választ adni az összegyűjtött adathalmaz) Hogyan viszonyulnak egymáshoz a tárolandó az adatok? (ezt majd az EK diagram készítésekor fog felszínre jönni)

A fenti kérdésekre adjuk meg a választ az egyed-kapcsolat diagram jelölésrendszer segítségével. A végleges EK diagramot kérjük ide beilleszteni.

Példa egy egyszerű EK diagramra:



3. Adatmodell bemutatása

Ezt a feladatot a normalizálásnak meg kell előznie. Az adatmodell majd a tervezés végére fog elkészülni, de azt ide, a második pontba kell bemutatni. Az adatmodellt majd az SQL Server Management Studio (SSMS) ból kell kivágni, és ide beilleszteni képként.

Figyeljünk arra, hogy minden egyed teljesen látszódjon, és igazítsuk majd el a táblákat úgy, hogy álló lapon is értelmezhető, olvasható legyen.

Féléves feladat sablon

4. Relációs sémák

Az összegyűjtött adatok szegmentálása relációsémákra. Egy gyártó esetén a relációs séma így nézhet ki:

Gyarto (gyarto_id, gyarto_nev, gyarto_cim, gyarto_tel, gyarto_email)

5. Adatmodell egyedeinek táblázatos bemutatása

Ide kérjük táblázatban bemutatni az egyedeket. Ahány egyed, annyi tábla. Nézzünk példát egy gyártóra

Gyártó

Tulajdonság neve	Adattípusa (hossza)	Kötelező-e megadni, vagy sem	Egy mondatos leírás.
gyarto_id	int	NOT NULL	Gyártó azonosítója
gyarto_nev	nvarchar(50)	NOT NULL	Gyártó neve

Féléves feladat sablon

6. Megszorítási szabályok

Ide kérjük táblázatban (az alábbi módon) bemutatni azon tulajdonságokat, melyekhez tárolás előtti validálás szükséges. Csak azokat a tulajdonságokat kell itt bemutatni, amihez tartozik megszorítási szabály.

Tulajdonság neve	Adattípusa (hossza)	Megszorítási szabály	Validálás szöveges értelmezése.
gyarto_id	int	[gyaro_id] > 1000	A számnak nagyobb-nak kell lennie 1000-nél
...

7. Elsődleges kulcsok

Ide kérjük felsorolni az összes tábla elsődleges kulcsát az alábbi forma szerint: tábla név – tulajdonság neve.

Pl:

- Gyártó – gyarto_ID
- -

8. Indexek

Ide kérjük felsorolni az összes tábla indexelt tulajdonságát az alábbi forma szerint: tábla név – indexel ellátott tulajdonság neve.

Pl:

- Gyártó – gyarto_nev
- -

Féléves feladat sablon

9. Alapértelmezett értékek

Ide kérjük felsorolni az összes tábla olyan tulajdonságát melynek rendszer által beállított alapértelmezett értéke van, az alábbi forma szerint:

tábla név – alapértelmezett értékkel ellátott tulajdonság – alapértelmezett érték

Pl:

- Gyártó – gyarto_nev – AUDI
- - -

10. Automatikus növekmények

Ide kérjük felsorolni az összes tábla olyan tulajdonságát melynek automatikus növekménnyel vannak ellátva az alábbi forma szerint:

tábla név – tulajdonság neve – AUTOINCREMENT(start_value, increment_value);

Pl:

- Táblanév - ID - IDENTITY(1000, 5)

11. GIU tervek

Ide kérjük bemutatni a legalább 3, max 6 képernyőképet, amit elképzeltetek az alkalmazásnak. A képeket pár mondatos leírással kell ellátni, hogy az milyen „menüpontnak” a felülete, és ott milyen beállítás végezhető el.

Határidők

- Téma regisztrációs határidő:

2024. szeptember 13.

- Kész féléves feladat elektronikus feltöltése (FÁJLFELTÖLTŐ ŪRLAPON):

PLUSZ VIZSGAPONTÉRT:

2024. október 20., 12:00-ig

VÉGLEGES LEADÁSI HATÁRIDŐ:

2024. október 31-ig., 12:00-ig



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

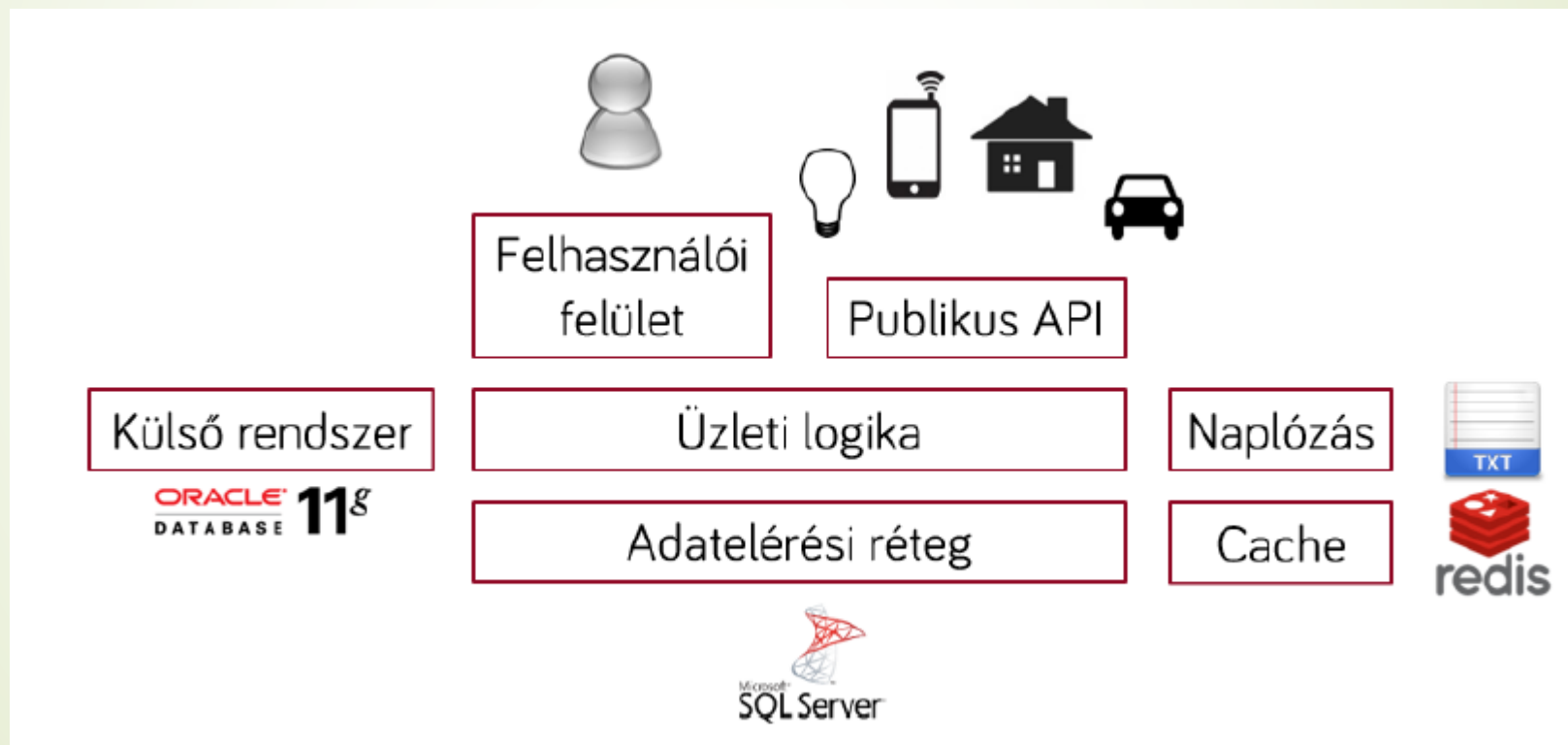
Adatelérési réteg

Adatbázis

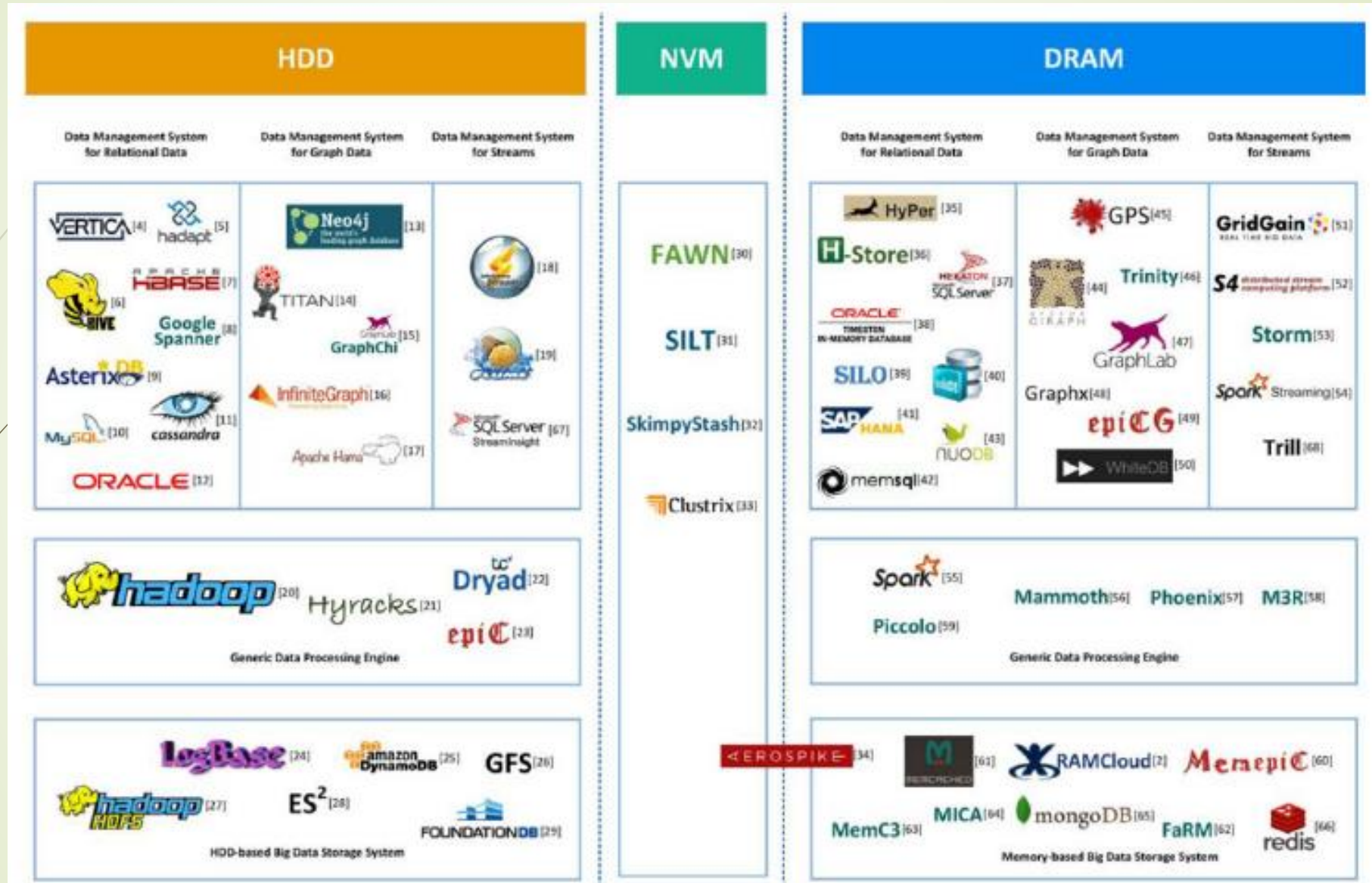
MI A MEGOLDÁS?

Adattárolás evolúciója

- Mini alkalmazások



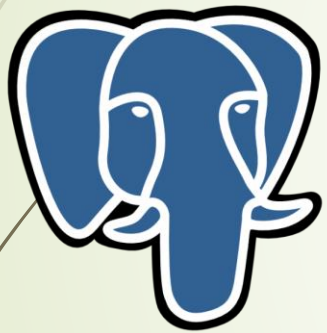
MI A MEGOLDÁS?



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).

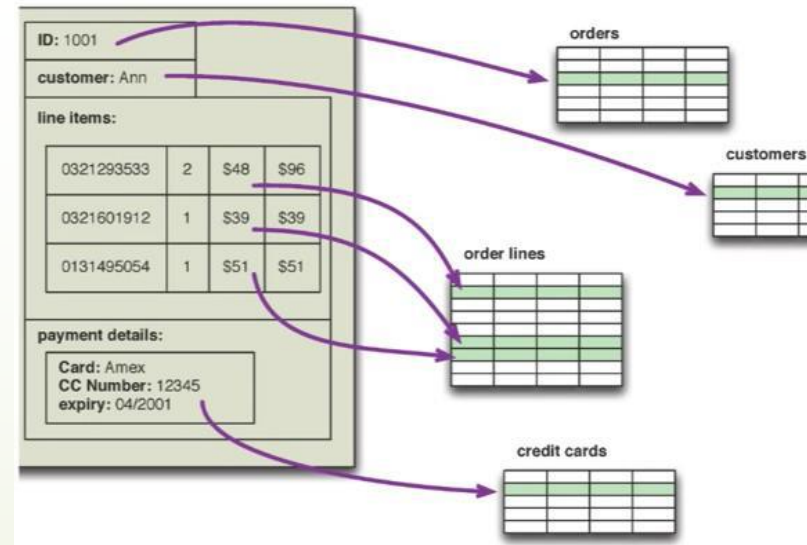
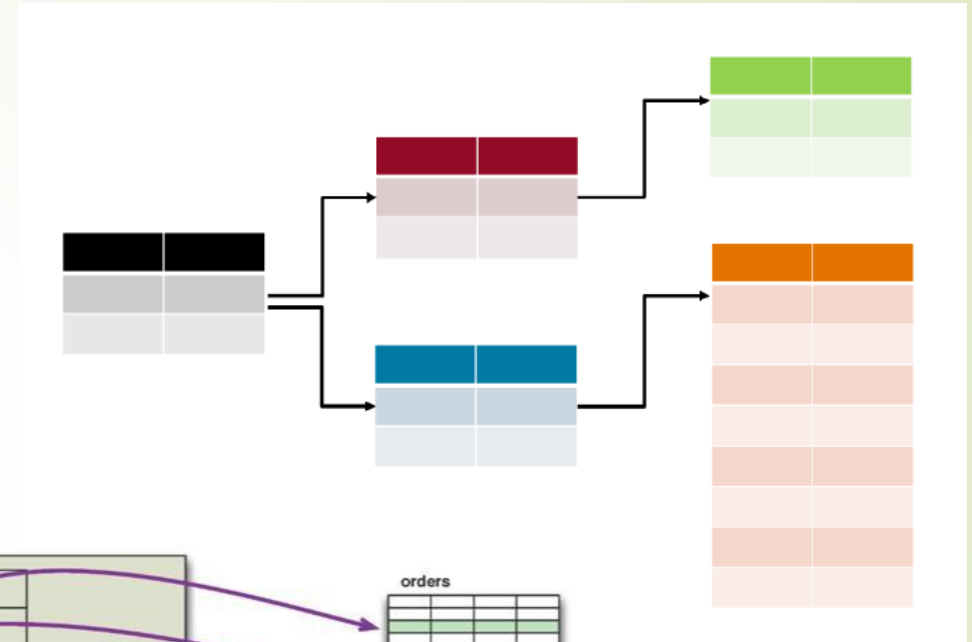
A táblák közötti kapcsolatokat közös tulajdonságokkal, indexeken keresztül valósítjuk meg.

- ▶ **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.

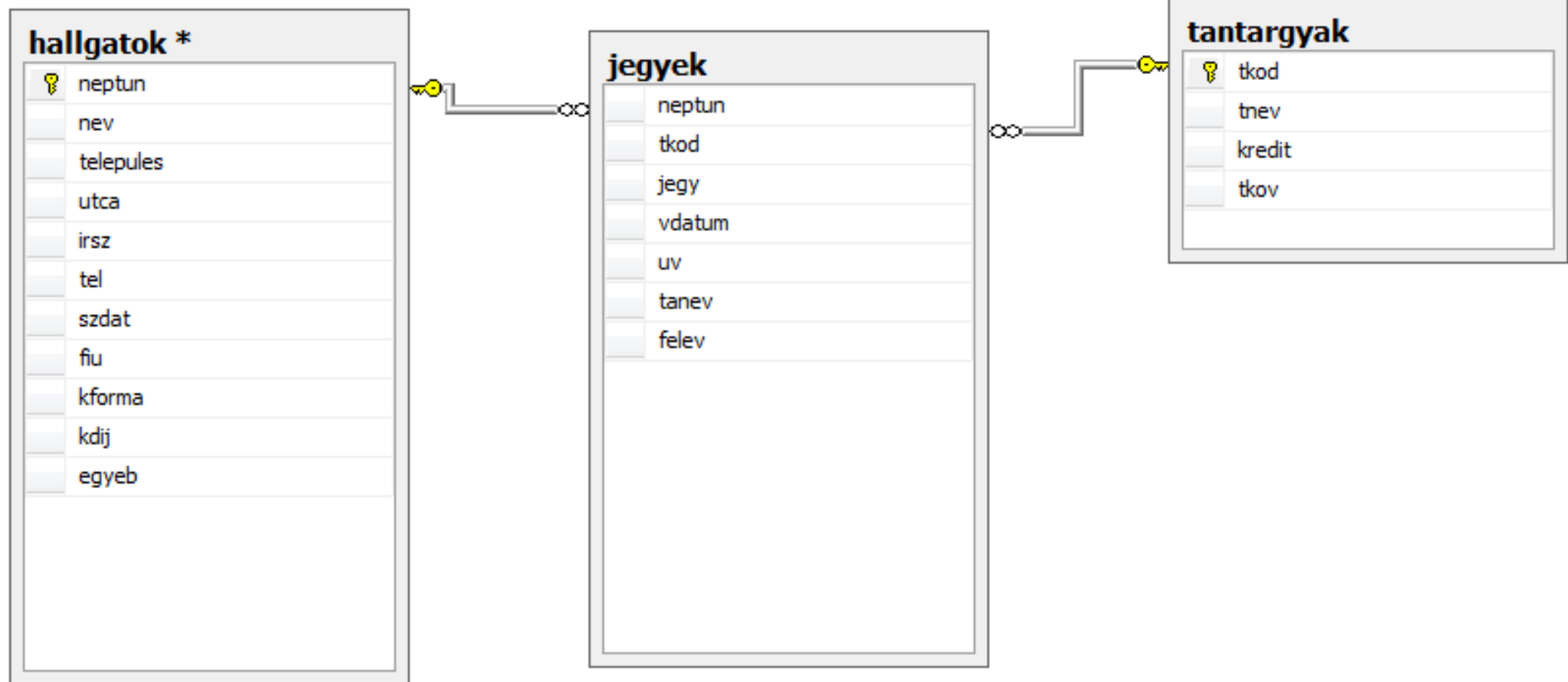
- **Egy-a-többhöz kapcsolat (1 : N) Pl.: Anya és gyermekei**

(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	- 9.223.372.036.854.775.807	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	922.337.203.685.477,5808	- +922.337.203.685.477,5807	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,



Köszönöm a figyelmet!

