

Előszó:

Ha nem érted meg először a Jt lényegét ne tégy szemrehányást magadnak.

Ne érezd magad rosszul,mert a számítástechnikai szakemberek, adatbáziskezeléssel foglalkozó részének is nagyon nehéz dió.

A nehézséget nem a tárgyi tudás hiánya, hanem a paradigmaváltás jelenti.

Egy bizonyos cél elérése érdekében be kell áldoznunk eddigi felfogásunkat,el kell engednünk bizonyos illúziót.

Ez a megszokás szakembereinek megoldhatatlan feladatot jelent a nem szakembereket pedig nem igazán hozza lázba. :))

Szívesen elküldöm a mai írásomat erről a témáról. Bence doktori tézisét fogja alkotni reményeink szerint.

Témája :hogyan és miért tároljuk el adatainkat egy 4 oszlopos táblázatba ?

A miértre az a felelet hogy egy darab 4 oszlopos táblát ,lehet automatizálni 100 oszloposat nem.

A mechanikus fizikai munka gépesítése, a mechanikus szellemi munka számítógéppel történő segítése után , a tudás közkinccsé tétele következik (reményeim szerint).

Ehhez (automatizálható tudás rögzítéshez) van egy ötletünk.

Ítt azt írom le ,hogy milyen lépésekkel jutottunk el ehhez ötlethez.

A lépések egymásutánja talán jobban követhető ,mintha az elején a végeredményt tálnánk.

Ha érdekel a téma ,akkor ezen a linken megtalálod azt a könyvet, amire később utalok.

<http://mek.oszk.hu/11100/11123/11123.pdf>

Az ID története

-Tizenvalahány évig úgy fejlesztettünk,hogy az összetett kulcsok értékei egyediséget kölcsönözzenek a táblasorainknak.

Eleve ezért és így szerveztük az összetett kulcsainkat.

Akkoriban nagyon sokan tettünk így.Pl az alábbi tábla leírja, hogy egy adott kft-ben kik fordultak már meg, valamilyen célból.

Akár egyszer fordult már meg akár 100- szor elég egyszer feljegyeznünk a nevét az alábbi táblában.

Kft_kód	személy_kód	megnevezés
1	1	Gipsz Jakab
2	2	Kis Péter
1	3	Nagy György
2	1	Gipsz Jakab
1	2	Kis Péter
2	3	Nagy György

Összetett kulcs =kft_kód + személy_kód egyediséget biztosít a táblán belül.

A gyakorlatban ez azt jelentette ,hogy egy kft_belül, egy adott személy (pl 1-es) csak egyszer szerepelhet.

1 +1 az összetett kulcsunknak (kft+személy) értéke egyedileg meghatározta az 1-es kft-ben előforduló Gipsz Jakab-ot (személy_kódja=1).

Az egyik barátunk tanácsára az egyediség biztosítására , kizárólag erre a célra alkalmazni kezdtünk egy ID mezőt .

Először nagy volt a kísértés, hogy ezt a mezőt másra is használjuk, de gyorsan leszoktunk róla.

Az ID minden táblába való bevezetése óriási találmánynak bizonyult. Annyira hasznos ez a mező, hogy most már nélkülözhetetlen a fejlesztésünkben és gondolkodásunkban egyaránt.

Az összetett kulcs egyediséget biztosító szerepe feledésbe merült. Az utolsó 10 fejlesztésünkbe már semmilyen szerepet nem kapott.

Nem kell összetett kulcsot szervezni többé a táblasoroknak, mert ott van az ID. A fenti táblánk ID-val kiegészítve így néz ki:

ID	Kft_kód	személy_kód	megnevezés
1	1	1	Gipsz Jakab
2	2	2	Kis Péter
3	1	3	Nagy György
4	2	1	Gipsz Jakab
5	1	2	Kis Péter
6	2	3	Nagy György

Itt már a táblasor egyediségét kizárólag az tábla ID-ja biztosítja. Egy ID-val egyszerűbb bánni mint 2 -3-4 összetett kulccsal.

Minél egyszerűbb valami, annál jobban átlátható, átadható, standardizálható, szabvánnyá és univerzálissá tehető.

PI egyetlen értékadással (6) kiolvasható a táblasorra jellemző kft_kód, személy_kód és a megnevezése is.

A Joker Tao gondolat nem kis mértékben az egyediségnek köszönheti létét.

A Joker Tao központi gondolata, hogy az egyediséget nem csak az egyes táblákban, hanem az alkalmazás egészére nézve, vagy akár az alkalmazások között is biztosítjuk. (PI ha a id=23 terméket azonosít, akkor az anyagok között a 23 biztos nem fog anyagot azonosítani)

Ha az egyik alkalmazásban már ID=23 terméként szerepelt, akkor egy másik alkalmazás is fel tudja használni ezt az összefüggést, ha szükségünk lesz valamikor erre. (Amire szükség lehet, arra előbb utobb szükség is lesz)

Ezért célszerű alaphól, zsigerből alkalmazások között is biztosítani az egyediséget, ha ezt megtehetjük.

Tulajdonképpen szabványokat építünk ki és erre építjük a rendszerünket.

Apró kitérő:

Az EDI (elektronikus adatcsere) ügyletág ennek a hiányára épít, pontosabban a vállaltok közötti szabvány, vagy egyeztetés, párbeszéd hiányára.

Az EDI azonosító párokkal ügylet: ami az egyik cégnek 1234 az a másiknak 4321. Ennyi és nem több a lényeg.

Ha az azonos dolgokat azonosnak jelöljük és így is hivatkozunk rá, akkor nem kell bonyolítani az életünket pl adatok cseréjével.

A JT célja az egyszerűsítés.

Az ATTRIBUTE története

Az alkalmazás fejlesztése során nagyon sokszor volt szükség kód-megnevezés párosra.

Ilyenkor a megnevezést (display) mutatjuk a felhasználónak, de a kódot, az értéket tároljuk a táblába.

Ezt a népszerű programozói szokást az Oracle-APEX objektumba öntötte és LOV-nak List of value-nak nevezte el. PI:

kód	megnevezés
1	Gipsz Jakab
2	Kis Péter
3.	Nagy György
4	Plósz István

Természetesen a programozónak semmi se elég és a kód-megnevezést párost mi is elkezdjük további mezőkkel hízlalni:

-jó lenne ide tenni,(kiegészíteni a táblát) a magyarázatot (note)
-jó lenne az érvényességét,aktív/inaktív elvégre tudni kellene, hogy él-e még az a megnevezés
-de jó lenne dátummal is ellátni ,mert annyira, de annyira szükség lenne rá
-én ezt a szerkezetet előszeretettel használtam dinamikus reportok kezelésére is,amihez hozzátartozott a nyomtató típusa, a report file_neve ,ami ugye már különbözött a report nevéől.

Az univerzális kód-name szerkezet variánsai alkalmazásaink tucatjaiba épültek be ilyen módon. PI Az alábbi példa egy iskola nyilvántartás. Ennek a példának az értékelését a mellékletben találjuk meg, de a fő irányvonalunkba ,csak áttételesen kapcsolódik ,ezért ezen a helyen csak példaként tekintünk rá.

A főcsoportok (hivatkozás nélküli sorok) megjelenítése				SQL
ID	ELEM	HIVATKOZAS	LISTASORREND	select id
1	Oktatók	0	-	,elem
2	Intézmények (telephelyek)	0	-	,hivatkozás
3	Csoportok	0	-	,listasorrend
4	Évfolyam	0	-	from szotar
5	Tanszak	0	-	where hivatkozás=0
A megjelenítése (genum és specifika) didaktikai sorrendben				
ID	ELEM	HIVATKOZAS	LISTASORREND	
1	Oktatók	0	-	figyeljük meg ,hogy a HIVATKOZAS értéke megegyezik az Oktatók id-jával
106	M. F. Csilla	1	1	
158	Sz. Tünde	1	2	
159	N. Zsuzsanna	1	3	
160	N. Jánosné	1	4	
2	Intézmények (telephelyek)	0	-	figyeljük meg ,hogy a HIVATKOZAS értéke megegyezik az Intézményi telephelyek id-jával
22	Vilmány	2	3	
25	Sátoraljaújhely	2	6	
107	Göncruszka	2	7	
135	Tiszakarád	2	9	
3	Csoportok	0	-	figyeljük meg ,hogy a HIVATKOZAS értéke megegyezik a csoportok id-jával
185	Kg-A/III-IV-109	3	1	

Az új alkalmazás fejlesztése mindig az unalomig ismert kód-megnevezés párosok, és a jól bevált variánsai megírásával, másolásával kezdődtek. Hú de jól meghaladtunk.

Miután hátradőlünk a nem létező karosszékünkbe, akkor kiderült hogy egy picit mégis csak kell variálni a más alkalmazásból átmertett szerkezeten.

Ez az ici-pici átdolgozás az esetek túlnyomó részében tovább búrjázott és a tekintélyes munka elvégzése után felmerült bennünk: "Nem jártunk volna jobban, ha másolás helyett tiszta lappal kezdünk ?"

Az egyik ilyen kód érték variánsnál merült fel, hogy mi lenne ,ha kód és a megnevezés közé egy típust is beillesztenénk.

Így a táblánk így nézett ki:

kód	típus	megnevezés
1	40=tanár	Gipsz Jakab
2	43=diák	Kis Péter
3.	41=vizsgáztató	Nagy György
4	40=tanár	Plósz István

Típus mező absztrahálásával,általánosításával született meg az univerzális tulajdonság az ATTRIBUTE .

Mi van ha szükségem van Gipsz Jakab telefonszámára , vagy Kis Péter email-jére ?

Gipsz-re és Kis-re így két sor fog vonatkozni.

Ettől a pillanattól kezdve a tulajdonság-bővítésnek egyetlen módja van: a **szabványosítható, automatizálható fizikai sor bővítés.**

Ekkor így módon módosult a táblánk:

ID	attribute	Value	Note
1	40=tanár	Gipsz Jakab	
1	11=tel_szám	06 20 151515	Vegyük észre, hogy a 11-es tulajdonság ugyanolyan ID után van ,mint Gipsz Jakab
2	43=diák	Kis Péter	
2	12=email	kis.peter@gmail.com	Vegyük észre, hogy a 12-es tulajdonság ugyanolyan ID után van ,mint Kis Péter
3.	41=vizsgáztató	Nagy György	
4	40=tanár	Plósz István	

A T_SEQUENCE és T_VALUE története

Kedvenc szakkönyvem Halassy Béla :Az adatbázistervezés alapjai és titkai könyve.

Nem túlzok, ha évtizedekig olvastam egy-egy fejezetét. Akkor kezdtem el olvasni, amikor már jó néhány alkalmazás fejlesztésén túl voltunk.

A könyv megismerése után jó sokáig úgy éreztem magam, mint az a bizonyos hályogkovács, akit mikor felvilágosítottak az emberi szemről többé már nem merte a beteg embertársai szemgolyójáról bugylibicskával levakarni a hályogot.

Ez a könyv ismertette meg az adatbázis oszthatatlan 3 egységével, ezek az egyed, tulajdonság, és érték. Ekkor jött a döbbenet ,mert:

-az univerzális ID bevezetésével és kiterjesztésével megvolt az egyed

-az univerzális attribute -vel megvolt az attribute

-a T_value ,akkor jelentse az univerzális értéket. Legyen annyira univerzális, mint az az adatbázis mező típus ,aminek definiáljuk.

További kérdések merültek fel :

1. Mi van azokkal a tulajdonságokkal, amelyeknek nem egy értékük van, hanem vektor típusúak.

Pl ha egy személynek nem egy telefonszáma van ? Egyik megoldás ha a telefonszám vektortulajdonságot növekvő indexel látjuk el.

ID	attribute	t_sequence	Value
1	40=tanár	1	Gipsz Jakab
1	11=tel_szám	1	06 20 151515
1	11=tel_szám	2	06 30 151515
1	11=tel_szám	3	06 70 151515
2	43=diák	1	Kis Péter
2	12=email	1	kis.peter@gmail.com
3.	41=vizsgáztató	1	Nagy György
4	40=tanár	1	Plósz István

Ebben a példában a tanár Gipsz Jakabnak 3 telefonszáma is van.

Az egyed és tulajdonság dialektikájának története

Ránézve a táblára zavart érzünk:

Mi a 40-es attribute igazi jelentése ? Gipsz Jakab-hoz, vagy a tanár-sághoz van köze ?

Az egyszerű,egyértelmű dolgok hívei vagyunk ezért ezt a bonyolult dolgot ketté kell választani:

Gipsz Jakab féle és Tanár féle tulajdonságra. Eképpen jártunk el:

40,41,43-es tulajdonságot meghagytuk a tanárság-nak,vizsgáztatónak és diáknak.

Bevezettük ezen felsorásnak a genum-ját és 30-assal elneveztük virtuális táblának

pl Mely excell táblába jegyeznénk az egyedünket: diák,tanár,vizsgáztató ?

20-as tulajdonságot kineveztük 'megnevezésnek', minden dolgok nevének bármiről is legyen szó.

ID	attribute	t_sequence	Value	note
1	20	1	Gipsz Jakab	A megnevezés-tulajdonság értéke:Gipsz Jakab
1	30	1	40	A virtuális tábla (30) tulajdonságának értéke 40 (tanár)
1	11=tel_szám	1	06 20 151515	
1	11=tel_szám	2	06 30 151515	
1	11=tel_szám	3	06 70 151515	
2	20	1	Kis Péter	
2	30	1	43	A virtuális tábla (30) tulajdonságának értéke 43 (diák)
2	12=email	1	kis.peter@gmail.com	
3.	20	1	Nagy György	
3.	30	1	41	A virtuális tábla (30) tulajdonságának értéke 41 (vizsgáztató)
4	20	1	Plósz István	
4	30	1	40	A virtuális tábla (30) tulajdonságának értéke 40 (tanár)

A megjegyzésbe leírtak miatt az alábbi sorokkal egészítjük ki a táblánkat

30	20	1	Virtuális tábla	id=30 . egyed megnevezés-tulajdonság értéke:Virtuális tábla
40	20	1	tanár	id=40 egyed megnevezés-tulajdonság értéke:tanár
41	20	1	vizsgáztató	id= 41 egyed megnevezés-tulajdonság értéke:vizsgáztató
43	20	1	diák	id= 43 egyed megnevezés-tulajdonság értéke:diák

Az egyed és tulajdonság dialektikáján azt értjük hogy vizsgált egyed-tulajdonság alany vagy állítmány szerepben van-e.

Ha Gipsz Jakab státuszát, virtuális táblához tartozását nézzük, akkor a **diák** tulajdonságként van jelen.

A **diák** egyedként van jelen, ha le akarjuk írni a 'diák virtuális tábla' tulajdonságát.

Pillanatnyilag csak a megnevezés tulajdonságát írtuk le.

PI leírhatnánk még azt a tulajdonságát is, ami megmutatja, hogy milyen mezőket tárolunk ebben a virtuális táblában.

Véletlenül tudjuk, hogy a megnevezésen kívül még telefon számokat is tárolhatunk.

Bizonyára ez nem teljes.

Az egyed és tulajdonság dialektikájának ábrázolása úgy történik, hogy az adott egyed-tulajdonságot hol egyedként (id-ként) hol tulajdonságként (attribute-ban) jelöljük.

Természetesen az email és telefon tulajdonságoknak is meg kell adni az egyed leírását a virtuális tulajdonsághoz hasonlóan:

ID	attribute	t_sequence	Value	note
1	20	1	Gipsz Jakab	A megnevezés-tulajdonság értéke:Gipsz Jakab
1	30	1	40	A virtuális tábla (30) tulajdonságának értéke 40 (tanár)
1	11	1	06 20 151515	
1	11	2	06 30 151515	
1	11	3	06 70 151515	
2	20	1	Kis Péter	
2	30	1	43	A virtuális tábla (30) tulajdonságának értéke 43 (diák)
2	12	1	kis.peter@gmail.com	
3	20	1	Nagy György	
3	30	1	41	A virtuális tábla (30) tulajdonságának értéke 41 (vizsgáztató)
4	20	1	Plósz István	
4	30	1	40	A virtuális tábla (30) tulajdonságának értéke 40 (tanár)
30	20	1	Virtuális tábla	id=30 . egyed megnevezés-tulajdonság értéke:Virtuális tábla
40	20	1	tanár	id=40 egyed megnevezés-tulajdonság értéke:tanár
41	20	1	vizsgáztató	id= 41 egyed megnevezés-tulajdonság értéke:vizsgáztató
43	20	1	diák	id= 43 egyed megnevezés-tulajdonság értéke:diák
11	20	1	telefonszám	id= 11 egyed megnevezés-tulajdonság értéke:telefonszám
12	20	1	email	id= 12 egyed megnevezés-tulajdonság értéke:email

A fenti táblaszerkezetben leírtakat nevezzük Joker-Tao (JT) strukturának és tartalomnak

A JT táblából is kiolvasható a következő definíciók:

Definíciók:

virtuális rekord:

Azonos Id értékkel rendelkező JT sorok összessége.

virtuális tábla:

Azon virtuális rekordok összessége, amelyeknek a "virtuális táblához tartozás" tulajdonság értéke megegyezik. (Plósz,Gipsz)

Néhány adat ábrázolási példa és elv, mint a való világ adat bázis vetülete:

1. Iskola nyilvántartás példa (táblázat).

2. Az ismeretlen tárolása, majd feldolgozása

(Mátyás Máté gondolata az adattól az információig):

3. Gondolatok a mozgás, a változás, a folyamat adatbázisban történő ábrázolásáról

Mellékletek:

1. Iskolai nyilvántartás példa

Egy dinamikusán bővíthető 2 szintű leírási jegyzékre találunk itt példát a "SZOTAR" oracle table-ban. Lásd "univerzális kód-name szerkezet variánsai" részt

Az iskola felépítését írja le az OKTATÓ, TELEPHELY, CSOPORT, ÉVFOLYAM, TANSZAK stb főcsoport és azokhoz tartozó konkrét értékét.

Ez megfelel az egyed genum és egyed specifika 1:N kapcsolatának, **ahol állandó és triviális a tulajdonság, nevezetesen a megnevezés** (itt elemnek neveztük el).

Az adott sor hivatkozás mezője a sor genum-ra utal.

A listasorrend pedig a genum által meghatározott csoport-tagok sorrendjét határozza meg.

Az ID egyediséget biztosít mind a genum egyedek, mind a specifika egyedek között. Szerkezetében JT strukturának felel meg ez a tábla..

Végtelen fa-szerkezet hozható létre ezzel a technikával ?

Elvileg igen, de gyakorlatilag miért állt meg a 2. szintnél az építkezés ?

A miért-re a vastagítással kiemelt mondatrész utal.

A tulajdonság variálás lehetősége szegényes, mert gyakorlatilag egyetlen 'megnevezésre' korlátozódik.

Ráadásul nagyvonalúan lemondunk a tulajdonság elemzéséről : vajon ez genum, vagy specifika tulajdonság ?

A tulajdonságot nem egyedhez hasonló módon kezeltük pl nem hoztuk (dinamikus) szintre, nem foglalkoztunk vele, mert megelégedtünk annak triviális jellegével.

Miért csak 2 szintig ábrázolta a fejlesztő az iskola nyilvántartását ?

A megfejtéséhez próbáljunk meg a fejlesztő fejével gondolkodni:

-1. Kérdés:

Az iskolába járó diákok jellemzőit miért nem itt tároljuk ?

-1. Válasz:

Mert lehetnek azonos nevű diákok, akiket további tulajdonságokkal kellene azonosítani.

-2. Kérdés: Melyik szintre tegyük a diákokat ?

2.1 A tanárokkal egy szintre , amikor minden évben vannak új és végző diákok ?

2.2 A tanárok, a tanszak, az évfolyam, vagy a csoport legyen a genus-juk ?

2. Válasz:

Nincs jó válasz erre a kérdésre és ez már a táblába kódolva volt.

Emlékezzünk csak erre a mondatra :

“... a tábla megfelel az egyed genus és egyed specifika 1:N kapcsolatának, **ahol állandó és triviális a tulajdonság , nevezetesen a megnevezés.** “

Azért nincs jó válasz a 2. kérdésre, mert állandó triviális, csak “megnevezés” tulajdonsággal nem lehet messzire jutni az adatbázis építésben.

Ezen nélkülözhetetlen tulajdonságon kívül mindig szükség van további tulajdonságokra pl azonosítás miatt név+ anyja neve+lakcím-re , vagy ha egy elemnek több genus-ja (mint ahogy, apja és anyja), akkor szükség van ezt kifejező modell-re, technikára, módszerre pl vektor_típusú tulajdonságra.

A kiindulási *tételek* (lásd vastagított mondatrész) előrevetítették az *ítéletet*.

Magyarul ez az ábrázolási mód (forma) , ezzel a gondolkodással (tartalom, logika) ennyire képes.

A mi ítéletünk:

Az egyed és a tulajdonság egyenrangú.

Dialektikája az egyik legfontosabb eszközünk az adatbázis építéséhez.

2. Az ismeretlen tárolása, majd feldolgozása (az adattól az információig):

A következő kísérletben egy vak embernek adnak egy eszközt, amelyet ha az ablakához tart különböző helyeken különböző számokat olvas fel neki, de megfigyeli, hogy ugyan azon a helyen mindig ugyan azt az számot hallja a géptől, ezeket rögzíti Joker Tao módon:

<i>ID</i>	<i>attribute</i>	<i>t_sequence</i>	<i>Value</i>	<i>(note)</i>
20	20	1	megnevezés	A megnevezés megnevezése
9	20	1	Ismeretlen hallott szám	9-es ID-val rendelkező valami, egyed amit hall a géptől, aminek a "megnevezése" [20] Ismeretlen hallott szám.

2	9	1	620
3	9	1	610
4	9	1	590
5	9	1	510
6	9	1	430
7	9	1	400

Miután rögzítette elmondják neki, hogy a gép egy spektrométer és a több rétegű üveg prizmaként felbontotta a természetes napfényt (vörös, narancssárga, sárga, zöld, kék, ibolya) és ezen elektromágneses hullámok, hullámhosszait hallotta nanométer mértékegységben, így a vak ember az ismeretlent tárolta és az elmondottak alapján a következő képen egészítette ki Joker Tao tábláját:

ID	attribute	t_sequence	Value	(note)
9	20	1	Ismeretlen hallott szám	9-es ID-val rendelkező valami, egyed amit hall a géptől, aminek a "megnevezése" [20] Ismeretlen hallott szám.
2	9	1	620	
3	9	1	610	
4	9	1	590	
5	9	1	510	
6	9	1	430	
7	9	1	400	
9	20	2	hullámhossz	A 9 ID-val rendelkező egyed 2. megnevezése hullámhossz (amit hallott a géptől)
90	20	1	mértékegység	A 90 ID egyed megnevezése "mértékegység"
9	90	1	[nm]	A hullámhossz mértékegysége nanométer
91	20	1	jel	
9	91	1	λ	A hullámhossz jele a "lambda" (λ)
2	20	1	Vörös	A 2 ID egyed megnevezése Vörös
3	20	1	Narancs sárga	
4	20	1	Sárga	
5	20	1	Zöld	
6	20	1	Kék	
7	20	1	Ibolya	
100	20	1	színek	A 100 ID egyed megnevezése színek
50	20	1	1:n	Egy az n-hez kapcsolat tulajdonság
100	50	1	2	A 2 ID egyed a színek csoportjába tartozik
100	50	2	3	
100	50	3	4	
100	50	4	5	
100	50	5	6	
100	50	6	7	

3. Gondolatok a mozgás, a változás, a folyamat és az N dimenzió adatbázisban történő ábrázolásáról

A mozgás, a változás, a folyamat ábrázolásáról (nem csak a JT) szerkezete

A témáról folyó beszélgetésnek több eddig nem tárgyalt aspektusát szeretném tolmácsolni nektek, ami rávilágít a téma genum jellegére is . :)))

Az adatbázisról, annak definíciójából a "leírás jelleg" domborodik ki nekem.

A való világ **vetületének** manifesztálódása a szubjektumon keresztül.

Egy tárgy vetülről nekem mindig annak árnyéka jut először eszembe.

Ezt gondolom a mozgás ábrázolásáról:

Ezt a vetületet (ami eredetiben általában 3D) én ,mint adatbázis-építő 2 dimenzióra képzem le.

Ez a két dimenzió az egyed és a tulajdonság.

A leképzés folyamata során (deriváláshoz hasonlóan ? kontrollt kérek) egy dimenziót "csonkolódott" az eredetihez képest.

Ezt akkor is igaznak hiszem, ha az adatbázis egyed megjelenését tekintve 3D (pl:hologram).

Itt nem a x,y,z tengellyel mérhető dimenzióról, annak csonkolásáról, hanem a valóság és annak adatbázis vetülete közötti különbségről beszélek, amit én egy dimenzió elvesztésével szemléltetek.

A való világban történő folyamat,mozgás, vagy változás adatbázisban való megjelenítéséről, én a következőket gondolom:

-Adatbázisban ezt is álló vetületként, screenshot szerű pillanatfelvételné szökták ábrázolni, annyi különbséggel ,hogy azt, egy objektumra jellemző idő tulajdonsággal látják el.

A Java programozásban egy objektumnak van adat és van metódus tulajdonsága. Érdekes megfigyelni, hogy mind a két fő objektum részt TULAJDONSÁG-gal jellemzik. Tehát a változásnak,mozgásnak, metódusnak is van tulajdonsága a JAVA szerint (is).

-Térjünk vissza a változás -hoz ennek adatbázisban történő megjelenítéséhez.

Én a változást hasonlóan, mint az (kép - mozgókép) viszonyaként tudom felfogni,értelmezni,ábrázolni és láttatni.

Mint ahogy a kártyalapokra rajzolt figura mozogni kezd a kártyalapokat pörgető ember kezébe. Úgyan így a mozgásként,változásként értékeljük a fázisok-képeit, az egyes fázisok adatbázis szerű leírásának sorozatát.

Az egyes képeken az állandó-ság mellett (azonos kártyalap, azonos nagyságú figura) a kisebb,nagyobb léptékű változást (pl emelkedő kéz) látjuk mozgásnak,cselekvésnek, történésnek (változásnak) .

Úgy is mondhatjuk, hogy a kártyalap-jellemzőitől eltávolodva ,alkottunk egy új tulajdonságot ,ami a kártyák közötti viszonyról,annak változásról, annak valamely tulajdonságáról szól. (nem az adott kártya-lapról)

Kiemelem ,ez a tulajdonság az új aspektus miatt jött létre.

Az "ismeretlen ismeretelmélet" adatbázisban történő ábrázolásáról:

-Én minden feladatban megelőlegezem az egyed ,a tulajdoság egyenrangúságát és annak dialektikáját. Mindezt azért teszem, mert eddig ez bejött nekem.

A feladat jellegéből adódóan ,ha ugyanannak az egyednek több időpillanatban történő alakja kerül jegyzésre, akkor én két módon tudnám a "látássérült ember a számokkal" példát elméletileg lejegyezni:

-I.Egyed oldaláról.

1.A vak ember egyedként lejegyzí amit hall. A hallott egyedet el is nevezheti és más egyéb tulajdonsággal is elláthatja (pl mit gondol róla)

2 lejegyzí,adott az info tár (tudat,agy) részeként a szín egyed genumot és piros,kék stb egyed specifikákat,mint létező,adott tudást,

3 A két egyedet (1.hang 2.szín egyedet) tulajdonság-párral (vélt egyed_id,valódi_egyed_id) összeköti, miután kiderült az igazság.

Kiegészítő egyezés :

- a) a vélt_id alapján megtudható, a vak által hallott, gondolt dolgok összége.
- b) a valódi_id alapján ezzel szemben megtudható, a tudástárban lerögzítettek.
- c) a két összekötött egyed kiegészül, azzal a tudással, hogy egy vak minek gondolta látás hiányában a pirost.

-II. Tulajdonság oldalról:

- 1. Adott a hallott szám-tulajdonsággal kifejezett egyed(ek) Egyenlőre nincs a név tulajdonsága megadva.
- 2. Adott a lejegyzett, létező tudás Szín-genum és a piros, kék specifika egyedek. (adott a név, és a frekvencia is)
- 3. Miután megtudtuk az igazságot azután :a neve nincs egyedhez újabb tulajdonság adunk hozzá ez a konkrét szín lesz.

Kiegészítő ,egyezség: az ismeretlen egyedek " hivatkozott névtulajdonsággal" rendelkeznek ,ez annyit tesz, hogy a konkrét név helyett a t_value= hivatkozás lesz pl 45

Konkrét adatmodell leírást az id története_3._mellékletében található.

Az egyed oldaláról történő megoldás előrevetíti az N dimenziójú adatok tárolásának lehetőségét.

Lásd melléklet.

4. melléklet