Bánné Varga Gabriella

FOGALMI ADATMODELLEZÉS ÉS RELÁCIÓS ADATBÁZIS-TERVEZÉS

Bánné Varga Gabriella

FOGALMI **ADAT**MODELLEZÉS ÉS RELÁCIÓS **ADAT**BÁZIS-TERVEZÉS

Elmélet és gyakorlat

TYPOTEX

© Bánné Varga Gabriella, Typotex, Budapest, 2022

Engedély nélkül semmilyen formában nem másolható!

A könyv megjelenését a Case Partners Kft. és a WEBváltó Kft. támogatta.

Lektorálta: Gollnhofer Gábor

ISBN 978 963 493 189 8

Kedves Olvasó!
Köszönjük, hogy kínálatunkból választott olvasnivalót!
Újabb kiadványainkról és akcióinkról a *www.typotex.hu*
és a *facebook.com/typotexkiado* oldalakon értesülhet.

Typotex Kiadó
Alapította Votisky Zsuzsa, 1989
A kiadó az 1795-ben alapított Magyar Könyvkiadók
és Könyvterjesztők Egyesülésének tagja.
Felelős kiadó: Németh Kinga
Felelős szerkesztő: Kótai Kata
Borítóterv és tördelés: Szalay Éva
Készült a Multiszolg Bt. nyomdájában
Felelős vezető: Kajtor Bálint

Tartalom

Köszönetnyilvánítás **11**
Előszó **13**

I. RÉSZ

FOGALMI ADATMODELLEZÉS **17**

1. fejezet **Bevezetés** **19**
Egy kis történelem 19
Az adatmodellezés fogalma, értelmezése 21
A fejlesztés 3 szintje 23
Barker-féle egyed-kapcsolat modellezés 26

2. fejezet **Egyedek, tulajdonságok, kapcsolatok** **27**
Egyedek 27
Tulajdonságok 28
Attribútum vagy egyed? 29
Származtatott adatok 29
Kapcsolatok 30
Kapcsolatok megnevezése 30
1:M kapcsolat 33
M:N kapcsolat 34
Feloldó kapcsolatok 34
1:1 kapcsolat 35
„Kapcsolatot megvalósító attribútumok” 35
Esettanulmány: Nobel-díjasok kezdő modell 36

3. fejezet **Modell haladó fogalmai** **43**
Egyedi azonosító 43
Nem áthelyezhető és korlátozott kapcsolat 48
Rekurzív kapcsolatok 49
Szupertípusegyedek 50
Íves kapcsolatok 53
Ortogonális szubtípusok 54
Esettanulmány: Nobel-díjasok továbbfejlesztett modell 59

4. fejezet **Egyéb üzleti szabályok és megszorítások ... 65**
Struktúra és megkötések ... 65
Procedurális vs. definitív ellenőrzés ... 66
Domain ... 66
Egyedek definiálása domain érvényes értékekre ... 67
Domain definiálása szövegesen ... 69
„Üres” értékek ... 70
Állapotátmenetek ... 70
Mátrix ... 70
Állapotátmeneti diagram ... 71
Egyedéletrajz ... 72
Grafikusan nem ábrázolható egyéb megszorítások ... 74
Peremfeltételek ... 76
Esettanulmány: Nobel-díjasok modell szöveges megszorítások ... 77
5. fejezet **Idő kezelése és változások modellezése ... 79**
Dátum attribútum és NAP egyed ... 79
Attribútumok változása, történet egyedek ... 80
Kapcsolatok változása ... 82
Szabvány kódok változásának követése ... 83
Bitemporális modellek ... 86
Esettanulmány: Nobel-díjasok modell, idő és változások figyelembevételével ... 88
6. fejezet **Fázis vége ... 91**
META modell ... 91
CASE eszközök ... 93
Egyed-kapcsolat diagramok elrendezése ... 94
Csapdák elkerülése ... 98
Legyezőcsapda ... 98
Szakadékcsapda ... 100
Módosítás helye ... 101

II. RÉSZ

RELÁCIÓS ADATBÁZIS-TERVEZÉS ... 103
7. fejezet **Adatbázisterv készítése ... 105**
Relációs adatbázis alapfogalmak ... 106
Táblák ... 106
Deklaratív megszorítások (Constraintek) ... 107
Adatbázis-objektumok grafikus ábrázolása ... 109
A transzformálás alapelve, filozófiája ... 112
Szabvány névkonvenciók alkalmazása ... 112
Alapvető transzformációk 1:N kapcsolattal ... 112
M:N kapcsolat ... 116

1:1 kapcsolat ... 116
Ívek ... 117
Általános ív megoldás ... 118
Szuper- és szubtípusok transzformálása ... 119
Szupertípustábla ... 120
Szubtípustáblák ... 121
Szuper- és szubtípustáblák együtt (ívmegoldás) ... 121
Domain kódtáblák megvalósításai ... 124
Mesterséges egyedi azonosítók ... 126
Üzleti megszorítások egyéb megoldási lehetőségei ... 127
Nem áthelyezhető kapcsolat ... 127
Korlátozott kapcsolat ... 127
1:M kapcsolatok kötelezősége az 1 oldalon ... 128
Állapotátmenetek ... 129
Audit adatok a tervben ... 130
És mi marad a fizikai adatbázis tervre? ... 130
Módosításokról még egyszer ... 131
Esettanulmány: Nobel-díjasok adatbázisterv ... 132
8. fejezet **Normalizálás ... 137**
A normalizálás és célja ... 137
1. normálforma értelmezési problémák ... 139
Normálformák ... 139
Normalizálás helye a modellalkotásban ... 145
Esettanulmány: Nobel-díjasok listájának normalizálása ... 148
9. fejezet **Denormalizálás ... 153**
Mit miért és mikor denormalizáljunk? ... 153
Leggyakoribb denormalizálási technikák ... 154
Származtatott adatok tárolása ... 154
Táblák teljes vagy részleges „összekapcsolása” ... 154
Detail adatok tárolása Master adatokkal ... 155
„Rövidre záró” kapcsolatok ... 156
Történeti táblákkal kapcsolatos technikák ... 156
Aktuális detail adat tárolása Master adatokkal ... 156
Időtartam vége oszlop bevezetése ... 157
Aktuális jelző ... 157
Domain kódtáblák kiiktatása ... 158
Táblák „összekapcsolása” ... 158
Érvényes értékek „bedrótozása” ... 158
Egyéb megoldások ... 159
Hierarchiaszint-mutató ... 159
Táblák „kapcsolása” kiegészítő FK oszloppal ... 159
Esettanulmány: Nobel-díjasok adatbázisterv denormalizálás ... 161

III. RÉSZ

MODELLEK AZ ADATTÁRHÁZBAN ... 163

10. fejezet **Adattárház és dimenzionális modell ... 165**
Adattárházak kialakulása ... 165
Az adattárház első szintje ... 167
Dimenzionális modellek az adatpiacokban ... 169
Csillag modell, dimenziók, tényadatok ... 170
Néhány bevett denormalizálás csillag sémában ... 171

11. fejezet **Data Vault modell ... 175**
Alapvető komponensek ... 175
3NF vs. Csillag modell vs. Data Vault modell ... 178

IV. RÉSZ

FELADATOK ... 183

12. fejezet **Feladatok és megoldások ... 185**
Feladat: Heti és napi időjárás-jelentés ... 187
Feladat: Tanfolyami katalógus ... 189
Feladat: Nyugta ... 191
Feladat: Házi könyvtár ... 194
Feladat: Szervezeti egységek ... 197
Feladat: Postai címek ... 198
Feladat: Recepttár ... 205
Továbbfejlesztés: Mértékegységek átváltása ... 210
Feladat: Étlap ... 212
Továbbfejlesztés/1: Menü bevezetése ... 214
Továbbfejlesztés/2: Helyettesítő tétel bevezetése ... 216
Továbbfejlesztés/3: Termékfajta menütételként ... 216
Adatbázis feltöltése ... 219
További megszorítások ... 222
Megszorítások a felhasználói felületen ... 223
Feladat: Pizza étlap normalizálása ... 225
Feladat: Munkatárs tábla normalizálása ... 227
Feladat: Termék tábla denormalizálása ... 228
Feladat: Nobel-díjasok összefüggő specifikáció ... 229

1. MELLÉKLET: Mi a címem? ... 233
A probléma ... 233
Honnan származnak a megváltoztatott adatok? ... 233
Címképzési és adatátvételi folyamat ... 235
Konklúzió ... 236

Címregiszter tervezése ... 237
Címregiszter tartalmi specifikáció ... 237
Értelmező rendelkezések ... 238
Címképzés ... 239
Lakóépületek típusai ... 239
Elméleti modell ... 242
Elméleti adatmodell attribútumokkal ... 244
Adatbázisterv ... 246
Címelemek domain tábláinak bevezetése ... 247
Épület típus és Állapot bevezetése ... 249
És a hiányzó üzleti logika ... 249
Címek szabványos megjelenítése ... 250
Tovább is van... ... 250

2. MELLÉKLET: Egyéb modellezési konvenciók ... 251
Chen-féle modell ... 252
Bachman-modell ... 253
Information Engineering/Martin ... 253

Név- és tárgymutató ... 255
Irodalomjegyzék ... 260

Köszönetnyilvánítás

Köszönetet szeretnék mondani elsősorban a könyv lektorának, Gollnhofer Gábornak, aki rengeteg munkája mellett vállalta a lektorálás időigényes feladatát. Nagy elméleti tudás és komoly gyakorlati tapasztalat birtokában tett javaslatai, megjegyzései sokat segítettek a könyv végleges verziójának kialakításában.

Köszönet illeti továbbá azokat a partnereket, akik megbíztak ilyen jellegű projektjeik megvalósításával; valamint az Oracle-nek és számos oktatási cégnek, hogy alkalmat adott a témában tanfolyamok megtartására Magyarországon és külföldön, és a hallgatóimnak, akik kérdéseikkel, felvetett problémáikkal szintén inspiráltak a könyv megírásában. Úgy gondolom, hogy az elmúlt 25 év, amikor párhuzamosan oktattam a legújabb tananyagokat, alkalmaztam az ott javasoltakat projektjeimben, majd a valós munka tapasztalatait visszaforgattam az oktatásba, szakmai életem leggyümölcsözőbb időszaka volt.

Előszó

A 2010-es évek elején fogalmazódott meg bennem, hogy sok-sok éves oktatói és fejlesztői munkám összefoglalásaként két könyvet megírjak. Kiadómnak is vázoltam a koncepciót: jó lenne egy részletes adatmodellezést tárgyaló kötet, és egy másik az adattárházak tervezéséről.

Direkt fogalmaztam most kicsit lazán, szeretném megmagyarázni az okát. Amikor 30 évvel ezelőtt elkezdtem foglalkozni adatmodellezéssel, adatbázis-tervezéssel, még nem léteztek adattárházak, és nem volt külön modellezési technikájuk. Ezért aztán lazán adatmodellezésről beszéltünk. Később, hogy megkülönböztessük a kétféle megközelítést, beszéltünk már operatív rendszerek modelljeiről, szakmai zsargonban 3NF (3. normálformájú) modellekről, illetve az adattárházi dimenzionális modellekről.

Mivel az említett időben adattárházakat terveztem, sok tapasztalat, gyakorlati eredmény állt rendelkezésemre, ezért úgy döntöttem, hogy ezzel a témával indítsunk. A könyv megjelenése óta eltelt időszakban ismét sok új tapasztalatot gyűjtöttem adattárház témában. Kollégáimnak szoktam mondani: akármennyi tapasztalatom van egy területen, egy új projekt mindig hoz új kihívásokat is a már ismerős rutinszerű megoldások mellett. Első gondolatom az volt, hogy egészítsük ki a 2012-ben kiadott, már régen elfogyott verzióját a könyvnek, és csináljunk egy új kiadást.

Ahogy ezt fontolgattam, eszembe jutott, hogy az adattárházaknál nemcsak az új dimenzionális modelleket tervezzük az elemzésekhez (ez a munka szebbik része), hanem a különböző forrásrendszerekből előbb létre kell hozni egy integrált céges/vállalati adatbázist, amelynek szerkezete az operatív rendszerekéhez hasonló, és finoman szólva is vegyes minőségű adatokat kapunk általában.

Az adattárházaknál így – és nem vagyok egyedül a tapasztalatommal – az adatokat illetően kétféle problémával szoktunk szembesülni:

– az adatszerkezetek zavarosak, redundánsak, rosszul strukturáltak, nem dokumentáltak és
– az adatok minősége sem megfelelő.

Most, hogy erről írok, szívesen nekiállnék furcsa adatanomáliákat bemutatni, de hát ez nem illik egy bevezetőbe, és különben is túl hosszú lenne (3. kötet?).

Az adattárházaknál cél minden adatot betölteni, és megvizsgálni, hogy az eleget tesz-e az üzleti elvárásoknak, s ha nem, regisztrálni a hiba tényét. Így történhetett, hogy az adattárházak mellékleteként előállt általában egy Hiba-adatpiac, amiből részletes elemzések készíthetők a forrásrendszerek minőségéről, az adatok hibáiról. Ezeket megpróbáltuk visszacsatolni a forrásrendszerek gazdáinak, általában

vegyes sikerrel. Voltak olyan hibák, melyek javításához ragaszkodtunk. Közben pedig mondogattuk: ezt is a forrásoldalon kellett volna megoldani…

Emellett az elmúlt években sokszor tapasztaltam új informatikai rendszerek használata során a felhasználói felület logikátlan voltát, valamint hogy megenged hibás adatfelvitelt. Amikor felhasználói felületről írok, akkor csak feltételezem, hogy azok valamilyen szinten tükrözik az adatbázis szerkezetét, mert azokba persze nincs belelátásom.

Így jutottam arra a felismerésre, hogy inkább az elsőnek tervezett kötettel kellene folytatni. Mire a tartalma összeállt, eljött a koronavírus-karantén, amikor a kényszert kihasználtam, és nekifogtam az írásnak. Az elhatározásomban megerősített egy valós életbeli saját problémám, amikor lakásunk címének furcsa, ráadásul nem egyértelmű változásáról értesültem. Próbáltam kielemezni az anomáliákat, nevezetesen, hogy ugyanannak az ingatlannak időben miért volt, illetve van négy különböző címe. Ebből született az 1. melléklet, amely körüljárja a problémát, és próbál megoldást bemutatni egy új cím modellel, ahol az üzleti logikán nagy hangsúly van.

Egy bekezdés erejéig vissza kell térnem az adatmodell kifejezésre, és már az Előszóba egy kis terminológiát becsempészni. Elsőre az *Adatmodellezés és relációs adatbázis-tervezés* címet akartam adni a könyvnek. Aztán rájöttem, hogy adatmodellezés alatt az irodalomban sokan mást és mást írnak le, értenek. Írnak logikai, fizikai, fogalmi modellezésről, elemzésről, relációs adatmodellről, és sok gyakorlati szakember fejében is maximum az adatbázis strukturális tervezése villan be a kifejezés hallatán.

A „legelején" szeretném kezdeni a könyvben, vagyis a fogalmi modellezéssel, amely során a szervezet vagy a vizsgált terület információigényét fogalmazzuk meg **függetlenül** az adatok tárolásától, hozzáférési mechanizmusoktól, a valóságnak megfelelő természetes fogalmakat használva. Egy ilyen modellt „vihetünk tovább", és átültethetjük az adatbázis logikai szerkezetébe, ahol már figyelembe veszünk technikai, hatékonysági, hozzáférési követelményeket. Bizonyos esetekben nincs semmi akadálya annak, hogy ez a két szerkezet (szinte) azonos legyen. Esetünkben ez a második szint a relációs adatbázis terve. A harmadik szint a tényleges fizikai adatbázis leírása. Ezzel nem foglalkozunk, itt már a konkrét adatbázis-kezelő tulajdonságai is szerepet játszanak. A modellek megfogalmazására többféle konvenció alakult ki az idők folyamán, és azok részletezettsége sem azonos szintű. Könyvünkben a Richard Barker által kidolgozott jelölésrendszert használjuk, melyet az *Oracle CASE*Method Entity Relationship Modelling* könyvében publikált először.

A könyv a fentiek alapján több részre tagolódik.

- Az I. rész foglalkozik az adatmodellezés kialakulásával, a fogalmi adatmodellezés céljaival, feladataival, a különböző szintű modellek értelmezésével, és mint a cím is mondja, részletesen magával a fogalmi adatmodellezéssel.
- A II. rész mutatja be a relációs adatbázis-tervezést, kezdve azon, hogy hogyan lehet átültetni a fogalmi modellt relációs adatbázistervbe. Megvizsgáljuk, hogyan lehet ellenőrizni a modell minőségét normalizálással, milyen lehetőségek állnak rendelkezésre, ha hatékonysági szempontok miatt el kell térnünk a „tiszta" 3. normálformájú szerkezetektől (denormalizálás). Az intuitív egyed-kapcsolat modellezéssel szembeállítjuk a relációs adatelemzés módszerét és a kétféle megközelítés eredményének összevetését.

- Az első két részben tárgyalt lépéseket, technikákat egy esettanulmányon visszük végig. A feladat a Nobel-díjasok nyilvántartásával foglalkozik. Minden fejezet után a feladathoz újabb részleteket hozunk nyilvánosságra (mint egy igazi projekt során), és módosítjuk a modellt az új ismereteknek megfelelően.
- A III. részben kitekintünk az adattárház-tervezési alapelvekre, röviden bemutatjuk a dimenzionális modellezés lényegét, majd áttekintjük a Data Vault technikát, amelyet az adattárházak építése során használhatunk a dimenzionális adatpiacok kialakítását megelőző adatbázisrétegben.
- A IV. rész feladatokat tartalmaz. A feladatok különböző fázisokhoz (szintekhez) tartozhatnak, és különböző technikák gyakorlására, bemutatására szolgálnak. Vannak közöttük olyanok is, amelyek a fogalmi modellt és az adatbázis tervét is tartalmazzák. Néhány feladatnál megmutatjuk az adatbázis feltöltését, illetve a deklaratív ellenőrzések érvényesítését adatbázisszinten és felhasználói felület esetén is.
- A Nobel-díjasok nyilvántartásának specifikációját itt egyben is közöljük, azzal a céllal, ha valaki a nulláról akarja a feladatot megoldani.
- Az 1. melléklet foglalkozik egy valós életbeli probléma bemutatásával és a lehetséges megoldással.
- A 2. mellékletben néhány elterjedt modellezési jelölésrendszert mutatunk be.

A könyvben végig hangsúlyosan kezeljük a fogalmi és adatbázis-struktúrák tervezése mellett az adatokra vonatkozó minden ismeret, megkötés megadását, majd megvalósítását. Ezzel szeretnénk az előzőekben felvetett adatminőség-kérdésekre reflektálni. Így számos olyan megoldást, technikát is bemutatunk, amelyek nem részei közvetlenül a modellezés bemutatott módszerének, de az elemzést, tervezést segítik, kiegészítik, főleg az adatminőség kérdésében. Ilyenek például az egyed-életrajzok, az állapotátmeneti diagramok, bitemporális modellek.

Szeretném hangsúlyozni, hogy mind az elméletet, mind a gyakorlatot, a feladatok megoldását nagyon fontosnak tartottuk. Remélem, a két aspektus kellően támogatja egymást végig a könyvben.

Végül el kell még mondanom, hogy első „rendszeremet" én is az adatszerkezetek kialakításával kezdtem, és körülbelül a harmadik adatbázis tervezése során alkalmaztam magától értetődően, módszeresen és könnyedén a könyvben ismertetett technikákat. Minden egyes új projekt ma is nagy kihívás, amely sok izgalmat, kalandot és komoly felfedezési folyamatot jelent.

Nagyon izgalmas látni, hogyan bomlanak ki a részletek, és kerülnek egymással összhangba. Ebben a grafikus ábrázolási technika rengeteget segít. Szoktam hangsúlyozni, hogy ne mi döntsünk a felhasználó helyett, hanem „ássunk minél mélyebbre", nyomozzunk kitartóan a tények megismerésében.

Kívánom, hogy olvasóim karanténtól és vírustól szabad körülmények között olvassák a könyvet. Szeretném, ha minél több modellezési, tervezési tanácsot alkalmaznának projektjeikben, és ne adják fel még akkor sem, ha ez esetleg más módszer, mint ahogyan eddig dolgoztak.

I. RÉSZ
FOGALMI ADATMODELLEZÉS

1. fejezet

Bevezetés

Egy kis történelem

Az adatbázisok és az adatmodellezés párhuzamosan fejlődtek az 1960-as évektől kezdődően.

Ekkorra már komolyabb információs rendszerek fejlesztését igényelték a felhasználók, és az eddigi kevés adat és kis tárolókapacitás helyett komolyabb adatmennyiségek kezelése és tárolása vált szükségessé. Az addig szokásos számítások és különböző feldolgozások után az adatok kerültek középpontba. Több elméleti modell keletkezett adatbázis-kezelők fejlesztésére. Ezekből három terjedt el: a hálós, a hierarchikus és a relációs adatbázisok. A 60-as években a hálós és hierarchikus megvalósítások voltak népszerűek, köztük például a Bachman által tervezett IDS (Integrated Data Store) hálós adatbázis, amelyet a GE színeiben fejlesztettek. Az IBM akkoriban a hierarchikus verzióra fókuszált, és kifejlesztették az IMS-t (Information Management System).

Bachman emellett kifejlesztette a Data Structure Diagramot, amely a rekordok szervezésének, szerkezetének grafikus reprezentálására szolgál, de még nem egyed-kapcsolat modell.

A 70-es évek komoly irányváltást hoztak. Edgar F. Codd publikálta a relációs adatbázis-kezelő modelljét. Mint az 1.1 ábra táblázatából is látható, ettől kezdve indult be a relációs adatbázis-kezelők fejlesztése és kereskedelmi forgalomba hozatala. Peter Chen hozta létre az egyed-kapcsolat modellezés első verzióját.

A fejlődés nem állt meg, hiszen azóta is megjelentek újabb alapokon nyugvó adatbázis-kezelők, mint a multidimenzionálisak vagy a NoSQL adatbázisok. De a hagyományos információs rendszereket a mai napig általában relációs adatbázisban valósítják meg.

Peter Chen, az egyed-kapcsolat modellezés atyja, a következőképpen határozta meg azt:

„Az egyed-kapcsolat modell azt a természetes nézőpontot fejezi ki, hogy a való világ egyedekből és kapcsolatokból áll. A modell a való világ fontos szemantikai információit mutatja."

Hangsúlyosan kimondja, hogy az ún. data structure diagram csak a rekordok struktúráját ábrázolja, így első próbálkozásnak tekinthető a témában, semmiképpen sem felel meg az egyed-kapcsolat modelljének.

Év	Termék/Publikáció	Adatbázis-kezelő	Modellezés
1960	IBM DBOMP	Hierarchikus	
1964	General Electric IDS (Integrated Data Store)	Hálós	
1966	CODASYL: szabvány	Hálós (szabvány)	
1968	IBM IMS (Information Management System)	Hierarchikus	
1969	Bachman Data Structure Diagram		Egyed-kapcsolat modell „elődje”
1970	Codd: A relational Model of data for large Shared Data Banks	Relációs adatbázis szabályok	
1973	Cullinane (később CA) IDMS	Hálós	
1975	ANSI SPARC-3 Architecture	Adatbázis architektúra szabvány	
1976	Chen: Entity Relationship Modelling		Egyed-kapcsolat modell alapok
1979	Oracle, Ingres	Relációs	
1981	Informix	Relációs	
1983	IBM DB2	Relációs	

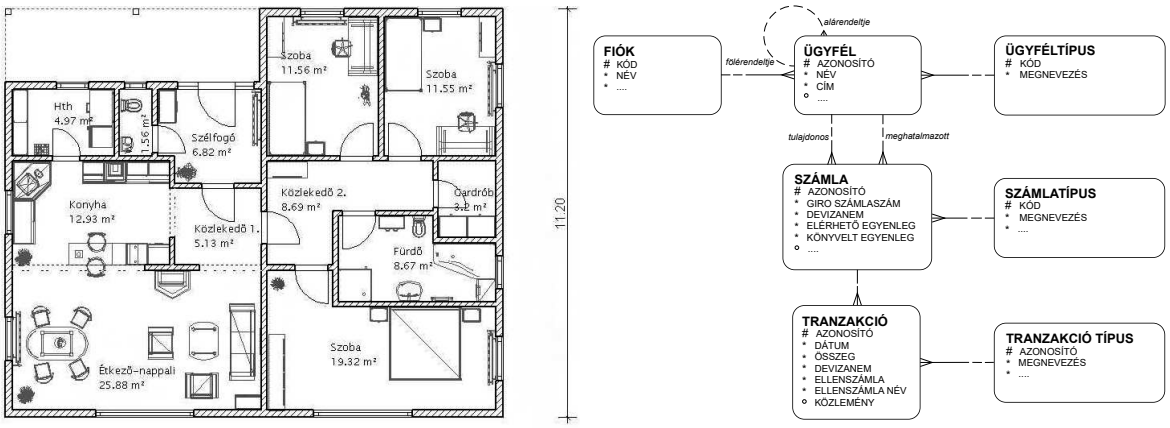
1.1 ábra: *Adatbázis-kezelő rendszerek és modellezés a 60-as évektől*

A témában későbbi kiemelkedő fejlesztők közül szeretném megemlíteni Richard Barkert, akinek 1990-ben adták ki *CASE Method: Entity Relationship Modelling* könyvét. Ebben sok éves gyakorlati tapasztalatait foglalta össze, amikor az Oracle-hoz csatlakozott. A könyv címében megjelenik a CASE Method kifejezés, ez két dolgot mutat: az Oracle már teljes fejlesztési módszertant hozott létre (Method), és a fejlesztési folyamatot számítógéppel támogatták (CASE). Ez egy újabb fontos állomás a témában, amikor szoftvereszközzel lehet leírni a modelleket, dokumentációt készíteni. Amint tudjuk, a CASE a Computer Aided Software Engineeringből összerakott betűszó.

Több neves cég alkotta meg ekkoriban a teljes fejlesztési életciklust lefedő módszertanát, ezek közül megemlítjük az SSADM-et (Structured System Analyses and Design Method), amely az Egyesült Királyságban de facto szabvány lett a kormányzati szektorban, és Magyarországon is használták bizonyos elemeit a 90-es években.

Az „Engineering” és a „Computer aided” kifejezések mind az építészeti, mind az információs rendszer tervezésénél jelen vannak. Ezt szeretnénk szemléltetni az alábbi ábrával. A számítógépes támogatás eszközei egyik oldalon a CASE eszközök, a másikon a CAD szoftverek.

Amit hangsúlyozni szeretnénk, hogy amíg az építészet területén teljesen triviális a különböző tervek készítése, engedélyeztetése, addig a szoftveres szakmában ez nem szabványos feladat. A 90-es években, amikor megjelentek az új módszertanok, CASE eszközök, úgy látszott, hogy ez bevonul a rendszerfejlesztésbe mint kötelező tevékenység, de mára ez visszafejlődött. Ha készítenek is dokumentációt, az sokszor visszafejtése a megvalósult rendszernek, célja, hogy formálisan teljesítsék a dokumentálást.



Tervdokumentációk:

Alaprajz
Szakági tervek

Adatmodell
Adatbázisterv

1.2 ábra: *Építészeti vs. adatbázis-tervezés*

Az adatmodellezés fogalma, értelmezése

Az informatikában az adatmodell kifejezést kétféle, nagyon eltérő értelemben használják.

1. Az adatmodell jelent **egyrészt fogalmi szintű** adatbázistervet, az adott alkalmazási környezet adatbázisának absztrakt tükörképét. Elvont a definíció egy kicsit? Más szavakkal kifejezve: fogalmi adatmodellnek nevezzük a jelenségeket, azok sajátosságait, viszonyait a **valóságnak megfelelő, természetes fogalmakban** való leírását. A leírás módja pedig az egyed-kapcsolat modell, így a fogalmi modellezést hívják egyed-kapcsolat modellezésnek is.

   Az adatmodellnek a szerkezeti vetületével mindenki tisztában van. De az adatmodellnek van egy másik dimenziója, mégpedig az adott alkalmazási környezet korlátrendszere. A korlátok a modell tényezőire és azok viszonyaira vonatkozó meghatározások, megkötések, kitételek. Vagyis a szerkezeti oldal mellett fontos a korlátrendszer, és ez is része a fogalmi adatmodellnek!

   Az adatbázis szerkezete önmagában is hordoz korlátokat. Például egy banki rendszerben egy tranzakciónak egy típusa lehet, és a típus értékének szerepelni kell a TRANZAKCIO_TIPUS

egyedtípus előfordulásai között. A szerkezeti modellben ilyen és hasonló jellegű korlátokat ki tudunk fejezni, de vannak olyan megkötések, amelyeket külön kell specifikálni és leírni.

És akkor elérkeztünk egy nagyon fontos szemponthoz, amelyet a bevezetőben már felvetettünk. Szeretnénk hangsúlyosan foglalkozni az efféle megkötésekkel, hogy minél jobb adatminőségű adatbázisokat tervezzünk. Az integritás kifejezést fogjuk erre a témára használni, amely az adatbázis megbízhatóságának, teljességének, épségének, érintetlenségének biztosítását fedi.

2. Egy másfajta értelmezés szerint [6] az adatmodell alapvetően adatok, illetve információ leírására szolgáló jelölésrendszer, amely 3 részből áll:

   - Az adatok struktúrája
   - Műveletek az adatokon
   - Megszorítások az adatokon

   A definíciót követően rögtön bevezetik például a relációs adatmodell kifejezést is. Itt az adatmodell az adatbázis „filozófiáját", típusát takarja, a konkrét alkalmazásoktól függetlenül leírják, hogy egy adatbázis-kezelő szerkezetét milyen tényezők alkotják, hogyan kell azokat megadni, és miképpen definiálhatjuk a korlátokat. Ezekre az adatmodell-koncepciókra jellemző, hogy nem fogalmi szintű elemekkel dolgoznak. A relációs modell reláció eleme se nem fizikai tárolási, se nem valós tényező, hanem logikai szintű elem. Ebbe a felfogásba az adatokon való műveleteket is beleértik, vagyis hogy milyen parancsokkal lehet az adatbázist kezelni, például ilyen az SQL nyelv.

	Építészeti tervezés	**Hagyományos téglaépítés**	**Könnyűszerkezetes építés (fa- vagy fémvázas)**
Struktúra	Alaprajzok	Téglafallal körülvett helyiségek elrendezése	Speciális falszerkezettel körülvett (pl. fa- és hőszigetelő rétegekkel kialakított, általában gipszkartonnal borított) helyiségek elrendezése
Megkötések	– Melyik helyiségből hova legyen direkt átjárás (ez már az alaprajzon is ábrázolt megkötés) – Helyiségek belmagassága	– Ajtók – Téglasorokból kialakítva	– Ajtók – Függőleges vázelemekből kialakítva
Műveletek, technológia ◆ **Építés** ◆ **Módosítások**		Lassú folyamat, menet közben lehet építéskor kisebb módosításokat végezni	Pontos tervet igényel, mert gyors a kivitelezés
		Falbontás, vésés nehézkes	Fal kivétele, bontása, új vezetékek beépítése egyszerű

1.3 ábra: *Struktúra – Műveletek – Megkötések házépítés során*

A **„Struktúra – Műveletek – Megkötések”** hármas értelmezéséhez ismét térjünk vissza az építészet témakörére, és nézzük, ott hogyan lehetne értelmezni a fenti szempontokat! A Fogalmi modellnek most az alaprajzi tervet feleltetjük meg, míg a különféle filozófiájú adatmodelleknek a hagyományos építésű és könnyűszerkezetes építés módját tekintsük. Természetesen csak egy-egy jellemzőt ragadtunk ki a szemléltetéshez és összehasonlításhoz.

Ebből is látszik, hogy sokkal kifejezőbb lenne az adatbázis-koncepció vagy filozófia kifejezés az adatmodell helyett. Találkoztam már olyan publikációval, amely abból indul ki, hogy az egyed-kapcsolat is egy ilyen adatmodell, kivéve, hogy nincsenek benne műveletek definiálva.

A fejlesztés 3 szintje

Ahogyan az átgondolt, „mérnöki” tervezés során több fázisban jutunk a tervhez, amelyet végül kiviteleznek, az adatbázis-tervezés folyamatában is több szinten történik a hasonló munka. További kifejezések, amelyekkel találkozhatunk: logikai és fizikai adatmodellek.

A szint itt fázist vagy lépést jelent, amelyeket egymás után kell végrehajtani. Mindegyik szinten más tényezőkre figyelünk. És itt használjuk a logikai és fizikai adatmodell fogalmakat.

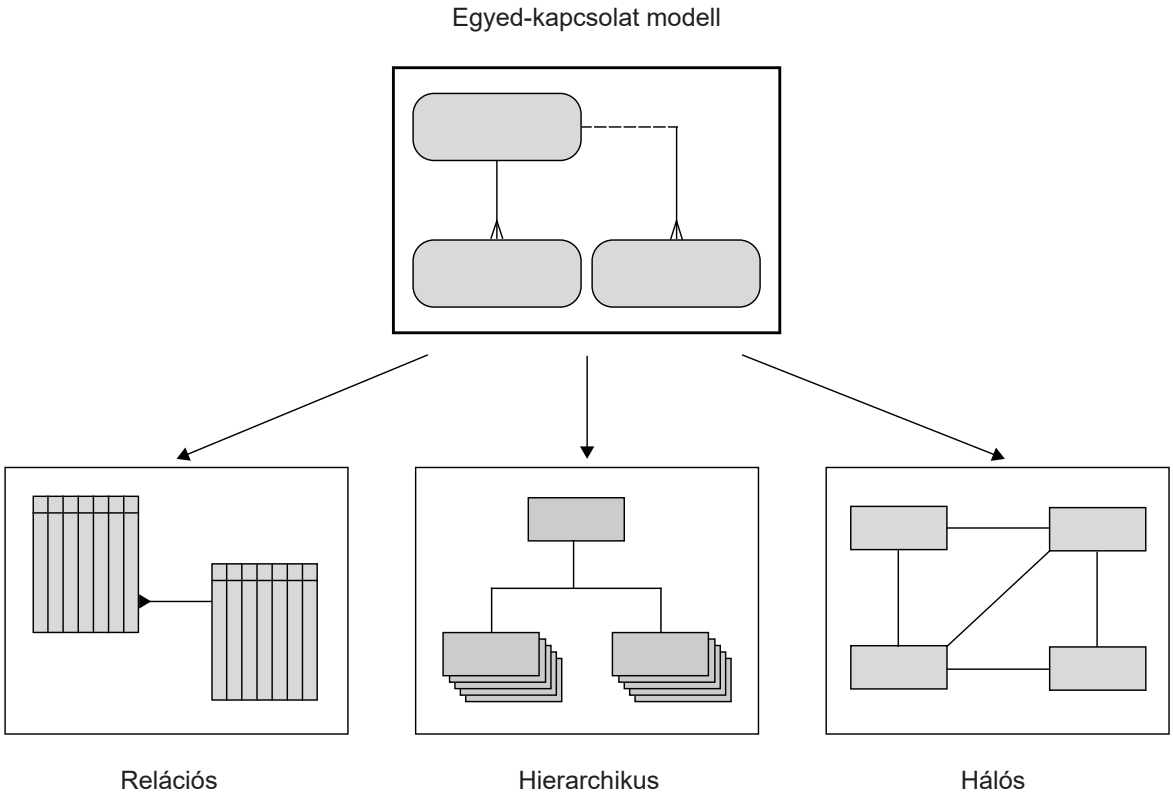
	Megnevezés	**Szinonima**	**Jellemzők, szempontok**
1. szint	**Fogalmi adatmodell**	Egyed-kapcsolat modell Koncepcionális modell	A jelenségek, azok sajátosságainak, viszonyainak a valóságnak megfelelő, természetes fogalmakban való leírása
2. szint	**Logikai adatmodell**	Adatbázisterv	Konkrét adatbázis-kezelő technikai architektúrájának, jellemzőinek figyelembevétele, hatékonysági szempontok beépítése
3. szint	**Fizikai adatbázisterv**		A konkrét hardver- és szoftverkörnyezetben történő implementálás terve

1.4 ábra: *Az adatbázis-tervezés szintjei*

Sajnos a terminológia még itt sem konzekvens. Olyan módszertanokban és CASE eszközökben, amelyek számára triviális a 3 szint megkülönböztetése, gyakran az 1. szintet nevezik logikai szintnek, a másodikat pedig például relációs modellnek (feltéve, hogy relációs környezetről van szó). A harmadik szint jelzője általában konzekvensen fizikai.

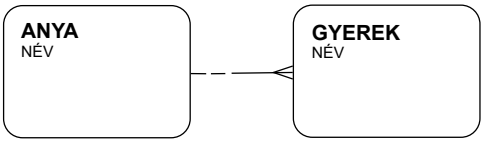
A következőkben szeretnénk bemutatni, hogy a fogalmi modell logikai szinten átültethető más és más adatbázis-kezelőbe, azaz az egyed-kapcsolat modell független a megvalósítás adatbázis-kezelőjétől. A relációs adatbázisban például idegenkulcs-oszlopokkal (és megszorítással) fejezzük ki az egyedek közötti kapcsolatot, míg a hierarchikus és hálós megoldásokban fizikai pointereket alkalmazunk a különféle módon összetartozó rekordok felfűzésére.

A hőskorban volt egy projektem, ahol a munka indulásakor nem lehetett tudni, hogy marad-e a jó öreg dBase „adatbázis-kezelő”, vagy megveszi a cég az Oracle DB-t. A fogalmi modell elkészült, teljesen függetlenül a megvalósítás adatbázisától, amely szerencsére Oracle lett.



1.5 ábra: *Fogalmi modellből tetszőleges adatbázis*

Nézzük az alábbi példát! Egy kicsit ugyan előreszaladunk, amikor azt mondjuk, hogy az ábrán egy fogalmimodell-részlet látható, amely azt mutatja, hogy egy ANYÁnak több GYEREKe lehet.



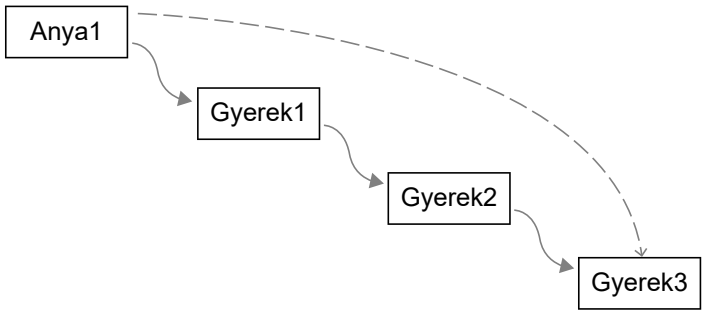
1.6 ábra: *Anya-gyerek modell*

Relációs adatbázisban a kapcsolat megvalósítása az ANYA nevének a GYEREK táblában történő elhelyezésével (idegenkulcs-oszlop) történik (az ANYA nevének az egyediségétől most tekintsünk el).

NÉV	ANYJA NEVE
Gyerek1	Anya1
Gyerek2	Anya1
Gyerek3	Anya2

1.7 ábra: *Anya-gyerek kapcsolat relációs adatbázisban*

Hierarchikus adatbázisban ezt a kapcsolatot sokkal bonyolultabban, pointerek segítségével oldhatjuk meg. Többfajta pointert használhatunk, a megoldás függ attól is, hogy direkt vagy szekvenciális jellegű elérésre van igény. Az alábbi ábrán pointerekkel fűzzük össze egy anya gyerekeit. Definiálhatunk például olyan pointert is, amely „visszafele" mutat, vagy az ábrán látható szaggatott vonallal ábrázolt pointer, amely az anya „utolsó" gyerekére mutat.



1.8 ábra: *Anya-gyerek kapcsolat hierarchikus adatbázisban*

Sokat gondolkodtam a könyv címén. Sokkal egyszerűbb cím lett volna az *Adatmodellezés és relációs adatbázis-tervezés* cím. De ekkor végiggondoltam az adatmodellezés imént bemutatott különböző értelmezéseit, és inkább a kissé hosszabb, de egyértelmű *Fogalmi adatmodellezés és relációs adatbázis-tervezés* kifejezést választottam.

Barker-féle egyed-kapcsolat modellezés

A mai napig sokan a Chen-féle grafikus egyed-kapcsolat modellt használják különböző példákban.

A 2. Mellékletben foglalkozunk a különféle jelölések bemutatásával. Én is csak oktatási célokra, bemutatásokra használtam a Chen-ábrázolásmódot, és nehezen tudnám elképzelni, hogy egy komoly rendszer modelljét ezzel rajzoljam. Lehet, hogy ez elfogultság? Az elmúlt közel 30 évben Richard Barker módszerét követtem [1].

Legfontosabb jellemzői elöljáróban:

- Természetesen fogalmi modellt készít, az „üzletre", nem az implementációra fókuszál.
- Grafikus úton dokumentálja a modell jelentős részét.
- Robusztus szintaxis segít a tények feltárásában és helyes leírásában.
- Jól olvasható és értelmezhető diagramok készíthetők. Természetesen rá kell szánni az időt a diagramok elrendezésére, átlátható kép kialakítására.
- Alapvetően OLTP, azaz online tranzakció feldolgozó rendszerek tervezésére szolgál, de használhatjuk egyéb területeken, ahol adatok tárolásáról van szó.
- A modellalkotás „felülről lefelé" haladó intuitív elemzéssel történik.
- A modell a létrehozandó adatbázis **integritására** fókuszál, az adatbázis legyen
  - korrekt vagyis hiteles,
  - konzisztens (azaz belső ellentmondásoktól mentes),
  - teljes körű,
  - mutassa mindig az aktuális (időszerű) adatokat.
- Számos részletet ad a modellhez, amelyet más módszertanok nem vizsgálnak.
- Tetszőleges típusú adatbázisban megvalósítható, legkönnyebben természetesen relációsban.

A könyvben nem leszünk teljesen egységesek az angol terminológia magyarítását illetően. A legfontosabb kifejezéseknek a magyar változatát használjuk, például entity helyett egyed (és nem entitás), unique identifier helyett az egyedi azonosítót használjuk, de vannak olyan kifejezések, amelyeknek nehéz magyar megfelelőt találni, ilyen például a Domain.

Az adatbázis-objektumok neveinél az angol és magyar bevett kifejezéseket is használjuk.