

# EGY FOLYÓIRAT ADATAIT TARTALMAZÓ ADATBÁZIS RELÁCIÓS MODELLJÉNEK ELKÉSZÍTÉSE

Hampel György

**Absztrakt:** A Jelenkori társadalmi és gazdasági folyamatok című folyóiratban az első megjelenés óta összesen 895 szerzőtől 482 cikk jelent meg. Annak érdekében, hogy az eddig publikált, illetve a későbbiekben megjelenő számokban található információk hatékony módon kinyerhetők legyenek, egy adatbázis létrehozásának ötlete merült fel. Az adatbázis elkészítéséhez szükséges volt egy magas szintű koncepcionális modell megtervezése, majd az alapján a logikai adatmodell elkészítése. A folyóirat kapcsán ez az egyed-kapcsolat modell megtervezését, majd relációs modellé konvertálását jelentette; ez utóbbi ennek az írásnak a tárgya. Az egyed-kapcsolat modell 3 egyedhalmazt, 39 attribútumot és 3 kapcsolatot tartalmazott. A konvertálás során ebből született 10 táblázat az egyedek adatainak tárolására – összesen 22 különböző mezővel (oszloppal) – és további 3 táblázat az egyedhalmazok közötti kapcsolatok biztosítására. Az így kialakított modell relációs adatbáziskezelőben megvalósítható és alkalmas a folyóirattal kapcsolatos információs igények kiszolgálására.

**Abstract:** A total of 482 articles by 895 authors have been published in Contemporary Social and Economic Processes since its first publication. In order to efficiently extract the information published so far and in future issues, the idea of creating a database has emerged. To create the database, it was necessary to design a high-level conceptual model and then build a logical data model based on it. In the context of the journal, this meant designing the entity-relationship model and then converting it to a relational model; the latter is the subject of this writing. The entity-relationship model consisted of 3 entity sets with 39 attributes, and 3 relationships. With the conversion, 10 tables were created to store the data of the entities – with a total of 22 different fields (columns) – and another 3 tables to ensure the relationships between the entity sets. The developed model can be created in a relational database manager and is suitable for serving information needs related to the journal.

*Kulcsszavak:* adatbázis, relációs adatmodell, Jelenkori társadalmi és gazdasági folyamatok folyóirat, Szegedi Tudományegyetem Mérnöki Kar

*Keywords:* database, relational data model, Journal of Contemporary Social and Economic Processes, University of Szeged Faculty of Engineering

## 1. Bevezetés

A Szegedi Tudományegyetem Mérnöki Karán a Mérnöki Menedzsment és Ökonómiai Intézete gondozásában 2006 óta jelenik meg tudományos publikációkat tartalmazó kiadvány, amely 2009 óta folyóiratként jelenik meg. Az azóta eltelt időben 2021. első feléig megjelent összesen 25 szám (a 2006. és 2007. évi évkönyvet is beleszámolva), a folyóirat jelenleg (2021. november) a 16. évfolyamánál tart. A 2019-es 3. szám már online ISSN számmal is rendelkezik (Hampel, 2020).

A folyóiratban első megjelenése óta számos, különböző tudományterületről származó cikk kapott helyet: a 16. évfolyam 1-2. számával bezárólag összesen 482 publikáció jelent meg, a szerzők száma összesen 895.

Felmerült, hogy létre kellene hozni egy adatbázist, amely könnyebbé teheti a folyóirat példányaiból származó lényeges információk kinyerését.

A megvalósítandó feladat első lépését a folyóirat adatait tartalmazó adatbázis megtervezése jelentette magas szintű modell (azaz egyed-kapcsolat modell) segítségével. Ennek részleteit tartalmazza Hampel (2020).

Jelen cikk az egyed-kapcsolat modell (az angol entity-relationship model elnevezésből a továbbiakban ER-modell) alapján készített logikai modell – azon belül a relációs adatmodell – leírását tartalmazza.

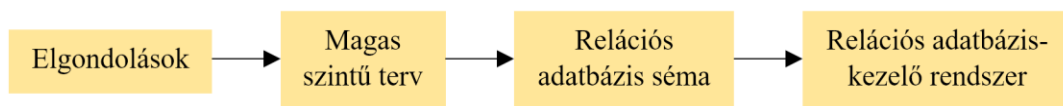
## 2. Egy adatbázis logikai modelljének elkészítése

### 2.1 A gondolattól a megvalósításig

Az adatbázis egy adott (szak)területet jellemző adatokból, valamint metaadatokból áll. A metaadatok az adatok típusát és az adatok közötti kapcsolatokat írják le. Az adatokat valamilyen rendező elv alapján csoportosítani kell annak érdekében, hogy azok adott szempontok szerint visszakereshetőek legyenek (Tímár et al., 1997).

Az adatbázis-modellezés és implementálás folyamatát mutatja be Ullman–Widom (2008) alapján az *1. ábra*. Korábbi cikkem (Hampel, 2000) a folyamat első két lépésével foglalkozott a folyóirat kapcsán, jelen cikk a 3. lépéshez kapcsolódóan született. Mind a négy lépés megjelenik Fabulya (2018a és 2018b) cikkeiben.

#### *1. ábra: Az adatbázis-modellezés és implementálás eljárása*



Forrás: Ullman–Widom (2008) alapján saját szerkesztés.

Az adatbázis mindig egy adatmodellhez kapcsolódik, azaz az adatbázis egy megvalósított adatmodell, amely a valódi adatokon kívül tartalmazza az adatok típusait és kapcsolatait leíró metaadatokat is (Hampel–Heves, 20019). Az adatmodellnek egyfelől elegendően robusztusnak kell lennie ahhoz, hogy megragadja a felhasználó figyelmét és megfeleljen az elvárásainak, ugyanakkor kellőképpen egyszerűnek, érthetőnek is kell lennie a felhasználó számára (Kroenke–Gray, 2006).

Amikor egy adatbázist magunk elé képzelünk, leginkább táblázato(ka)t látunk tele adatokkal. Talán innen jöhet a felhasználóktól néha hallható „minek az adatbázis-kezelő, jó lesz a táblázatkezelő is...”. Mindenesetre, ha egyetlen táblázatként képzeljük el az adatbázist, azzal már nagyjából tisztában vagyunk az egy táblázatból álló, egyszerű relációs modell felépítésével (Tímár et al., 1997): a táblázat oszlopokból (mezőkből) és sorokból (rekordokból) áll, tetején az oszlopok jelentését hordozó fejléccel (mezőnevek).

Vannak esetek, amikor egyetlen táblázatban nem tárolható, nem kezelhető hatékonyan az adatok tömege, mert az felesleges ismétlődéseket (redundanciát) eredményezhet, ami különböző bővítési, módosítási és törlési nehézségeket, problémákat (anomáliákat) okozhat (Tímár et al., 1997; Halassy, 2000). Az anomáliák elkerülése érdekében a táblát fel kell bontani több, egymással kapcsolatban

álló táblára különböző, ún. normalizálási szabályok felhasználásával (Halassy, 2000; Badia–Lemire, 2011). Az így létrejövő táblák hatékony kezelésére lettek kitalálva a relációs adatbázis-kezelők. (A Microsoft Excel táblázatkezelő Microsoft Power Pivot bővítménye ugyan már képes táblákat összekapcsolni és ezekből lekérdezések eredményét tartalmazó eredménytáblákat készíteni, de hatékonysága még elmarad a relációs adatbázis-kezelőktől.)

## 2.2 A relációs modell

Logikai adatmodellből többféle is kialakult a számítógépes adatbázis-kezelés kezdete óta, így létezik például hierarchikus, hálós, objektum-orientált, illetve a relációs adatmodell (Tímár et al., 1997; Silberschatz et al., 1996; Szabó, 2013). Ezek közül az utóbbi, a relációs modell terjedt el leginkább, az egyszerűsége és matematikai megalapozottsága következtében. Egy adatmodellhez kell tartozzon egy arra épülő adatbázis-kezelő is, így a relációs adatbázis-kezelők a relációs adatmodellre épülnek.

A relációs modell táblázatok rendszeréből épül fel. A táblázatok közötti logikai kapcsolat olyan tulajdonságok (mezők) segítségével hozható létre, amelyek tartalma mindkét kapcsolódó táblában megtalálható.

A relációs modell tábláinak felépítése a következő: A táblázatot olyan egyedi névvel kell ellátni, amely megkülönbözteti azt a többi táblázattól. A táblázat fejléce az egyedi oszlopneveket tartalmazza; ezek az oszlopnevek a tulajdonságok (attribútumok) azonosítói. Egy oszlopot mezőnek (ill. a fejléctet mezőnévnek) szokás nevezni. A táblázat egy-egy – fejléctől különböző sora egy rekord (vagy tuple). Az oszlopok száma a táblázat fokszáma, a sorok számát kardinalitásnak nevezzük. Az oszlopok, illetve a sorok sorrendje (a fejléctet kivéve) közömbös, azok tetszőlegesen felcserélhetők, a sorrendnek nincs kitüntetett szerepe. Azt a kiemelt tulajdonságot, amely alapján a táblázat egyes rekordjai egyértelműen megkülönböztethetők egymástól – elsődleges kulcsnak nevezzük – aláhúzással jelöljük. Ha több erre alkalmas mező is van, akkor egyet kiválasztunk elsődleges kulcsnak, a többi pedig alternatív kulcs. Előfordulhat, hogy több mező együttese alkot egy elsődleges kulcsot, ilyenkor összetett kulcsról beszélünk. A kulcsban szereplő mezők számának minimálisnak kell lennie. Az idegen kulccsal pedig hivatkozhatunk egy másik, hivatkozó táblázattal logikai kapcsolatban álló táblázatra (Tímár et al., 1997).

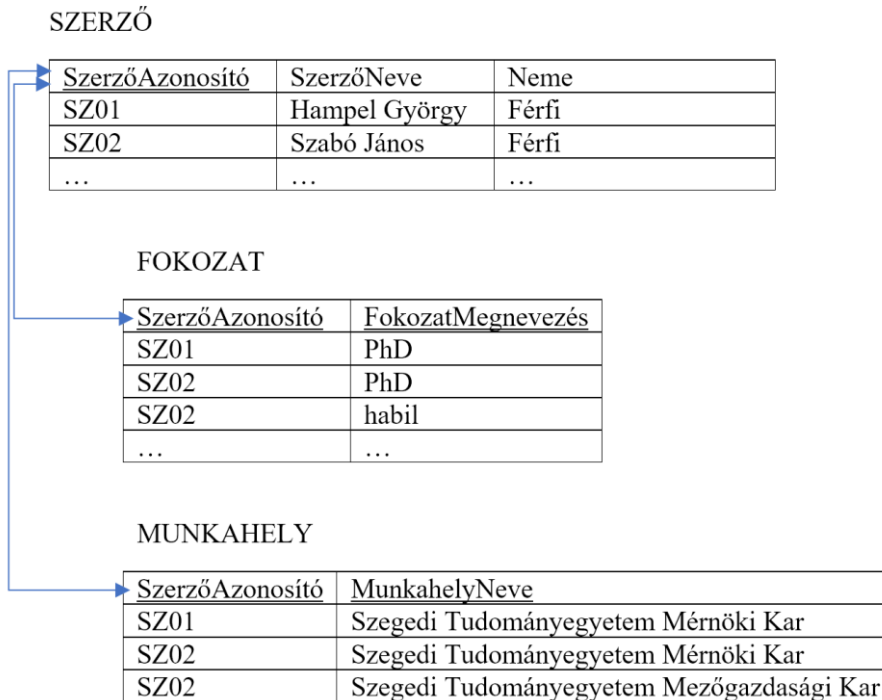
A modellezés során segítségünkre van a normalizálás is, amely táblázat-szétbontó műveletek sorozatából áll. A normalizáció során egy – sok esetben egyetlen (0. normál formában, 0NF-ben lévő) táblázatból álló modelltől – eljuthatunk több táblázatból álló, áttekinthető és a kiinduló állapotnál kisebb tárigényű (általában 3. normál formában, 3NF-ben lévő) relációs modellhez, amely többé-kevésbe mentes a redundanciától, illetve teljesen anomáliamentes (Halassy, 2000).

Az elkészült és adatokkal feltöltött táblázatokból relációs és egyéb kiegészítő műveletek segítségével nyerhetjük ki a kívánt információkat (Silberschatz et al., 2020).

A 2. ábrán példaként látható a folyóirat adatbázisának egy részlete relációs sémával leírva, valamint minta adatokkal táblázatos formában.

### 2. ábra: Relációs modell példa adatokkal

SZERZŐ (SzerzőAzonosító, SzerzőNeve, Neme)  
 FOKOZAT (SzerzőAzonosító, FokozatMegnevezés)  
 MUNKAHELY (SzerzőAzonosító, MunkahelyNeve)



A táblázatok közötti logikai kapcsolatokat nyilak jelölik.

Forrás: mintaadatokból saját szerkesztés.

### 2.3 Az ER-modell átalakítása relációs modellbe

Amennyiben a koncepcionális modellalkotás ER-modell segítségével történik, szükséges annak átalakítása relációs modellre. Ennek az oka, hogy bár az ER-modell könnyen megtanulható és használható eszköz, nem létezik hozzá olyan adatbázis-kezelő, amely ezt a modellt használná. Az átalakítás során használható fő szabályok a következők (Tímár et al., 1997; Dey et al., 1999, Kovács, 2004; Ullman–Widom, 2008):

- Egyed típus leképezése: Minden egyed típusnak egy relációt, azaz egy táblázatot feleltetünk meg.
- Tulajdonságok leképezése: Egyszerű tulajdonságok esetén az ER-modell attribútumai a táblázatok mezői lesznek; a mezők elnevezése megegyezik az ER-modellből származó attribútumnevekkel. Összetett tulajdonságoknál csak a résztulajdonságokat vesszük fel a táblázatba. A többértékű

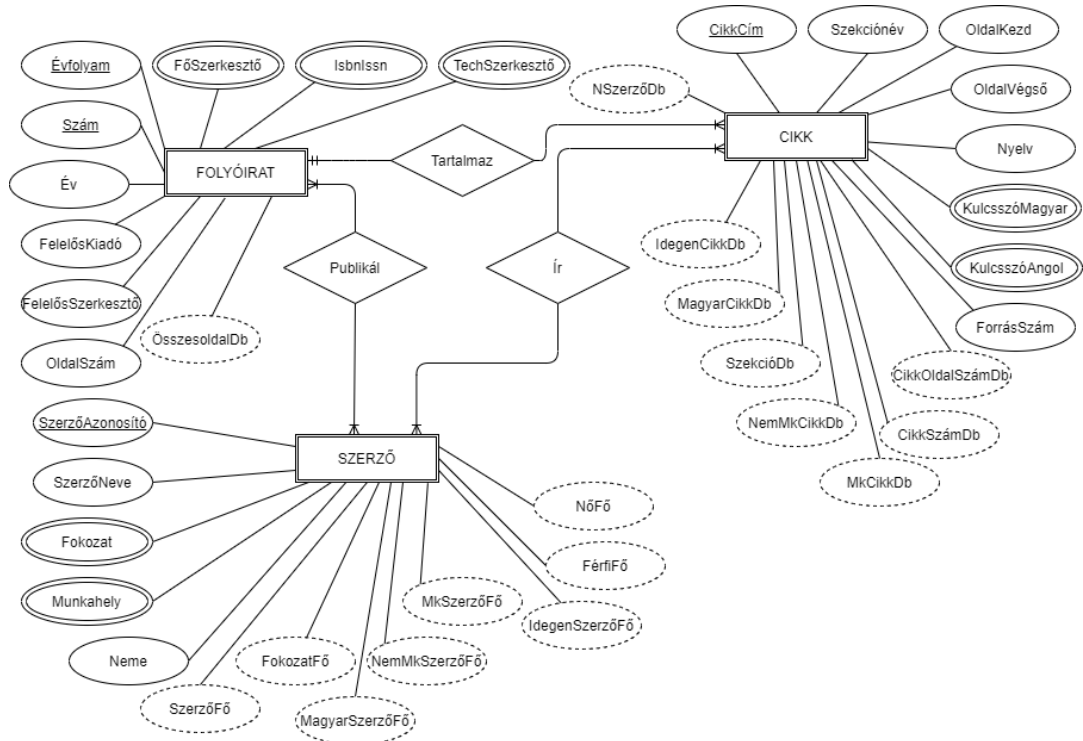
tulajdonságokat – mivel a táblázat egy-egy cellája csak egy elemi értéket tartalmazhat – kiemeljük egy külön táblába és idegen kulcs segítségével az eredeti táblázathoz kapcsoljuk. A leszármaztatott tulajdonságokat nem vesszük fel a modellbe, így nem szerepelnek egyik táblázatban sem, mivel más mezők adataiból kiszámíthatók.

- Kapcsolatok leképezése: Az átalakítást a kapcsolat típusa alapvetően meghatározza; a lehetőségek a következők: (1) Az egyik táblázat elsődleges kulcsát beágyazzuk a másik táblázatba idegen kulcsként és az mindegy, hogy melyik táblázatból melyik táblázatba – ez 1:1 típusú kapcsolatnál alkalmazható; (2) Az egy oldali táblázat elsődleges kulcsát beágyazzuk a másik – azaz több – oldali táblázatba idegen kulcsként, – ez 1:N típusú kapcsolatnál alkalmazható; (3) Létrehozunk külön, egy kapcsolatot biztosító táblázatot a két összekapcsolandó táblázat közé, amely tartalmazza a kapcsolt táblák elsődleges kulcsát – kizárólag ez a megoldás alkalmazható N:M kapcsolatnál és n-ed fokú kapcsolatnál, vagy ha a kapcsolatnak van tulajdonsága, de a módszer alkalmazható 1:1 és 1:N kapcsolat esetén is.

### 3. A folyóirat relációs modellje

A koncepcionális modell ER-modell formájában már rendelkezésre állt (lásd: 3. ábra). Ebből kiindulva a 2.3 alfejezetben ismertetett, átalakításra vonatkozó szabályok figyelembevételével történt a táblázatok létrehozása.

3. ábra: A folyóirat ER-modellje



Forrás: Hampel (2020).

Az ER-modell alapján készült relációs séma leírásánál a táblázatok neve nagybetűvel, majd zárójelk között a tulajdonságok, azaz a mezőnevek (oszlopnevek) szerepelnek az elsődleges kulcsok aláhúzásával:

1. A Folyóirat egyedtípus és tulajdonságai: Ez az egyedtípus a következő 4 táblázatra bontható: FOLYÓIRAT, amely tartalmazza az egyedtípus kulcstulajdonságát – amely 2 attribútumból álló összetett kulcs (évfolyam, szám) –, az egyszerű tulajdonságokat (év, felelős kiadó, felelős szerkesztő, oldalszám). A FŐSZERKESZTŐ, ISBNISSN, TECHSZERKESZTŐ táblázatok a FOLYÓIRAT táblázattal kapcsolatot biztosító egyedi azonosítókat (a célszerűség érdekében nevük megegyezik a FOLYÓIRAT táblázatban található elnevezéssel), valamint a főszerkesztők nevét, vagy ISBN és ISSN számokat, illetve a technikai szerkesztők nevét tartalmazza. A séma részletesen:

- FOLYÓIRAT (Évfolyam, Szám, Év, FelelősKiadó, FelelősSzerkesztő, Oldalszám)
- FŐSZERKESZTŐ (Évfolyam, Szám, FőszerkesztőNév)
- ISBNISSN (Évfolyam, Szám, IsbnIssnSzám)
- TECHSZERKESZTŐ (Évfolyam, Szám, TechSzerkesztőNév)

Az ER-modell tartalmaz a Folyóirat egyedtípusban egy leszármaztatott tulajdonságot, amely nem kerül bele egyik táblázatba sem, mivel kiszámítható az oldalszám adatokból: ÖsszesOldalDb.

2. A Szerző egyedtípus és tulajdonságai: Az ER-modell Szerző egyedtípusa a következő táblázatokra alakítható át: A SZERZŐ táblázat tartalmazza a szerzők azonosítására alkalmas elsődleges kulcsot (SzerzőAzonosító), a szerzők nevét és nemét egyszerű tulajdonságként (SzerzőNeve, Neme). A többértékű tulajdonságok a FOKOZAT és MUNKAHELY táblákba kerültek. Ez utóbbiak a SZERZŐ táblázattal kapcsolatot biztosító idegen kulcsot (SzerzőAzonosító), valamint a szerzők tudományos fokozatainak megnevezését, ill. munkahelyeik mengevezését tárolják. A séma részletesen:

- SZERZŐ (SzerzőAzonosító, SzerzőNeve, Neme)
- FOKOZAT (SzerzőAzonosító, FokozatMegnevezés)
- MUNKAHELY (SzerzőAzonosító, MunkahelyNeve)

Az ER-modell ebben az esetben is jelöl a Szerző egyedtípusban több leszármaztatott tulajdonságot, amelyeket – mivel más tulajdonságokból kiszámíthatók – a táblázat nem fogja tartalmazni, így azok nem kerülnek bele a modellbe; ezek az attribútumok a következők: NőFő, FérfiFő, SzerzőFő, IdegenSzerzőFő, MkSzerzőFő, NemMkSzerzőFő, MagyarSzerzőFő, FokozatFő.

3. A Cikk egyedtípus és tulajdonságai: A koncepcionális modellben szereplő Cikk egyedtípusból a következő táblázatok hozhatók létre a relációs modellben: A CIKK táblázat tartalmazza a cikkek egyedi azonosítóját (elsődleges kulcs: CikkCím) és a cikkhez tartozó egyszerű tulajdonságokat (SzekcióNév, OldalKezd, OldalVégső, Nyelv, ForrásSzám). A

KULCSSZÓMAGYAR és KULCSZÓANGOL táblázatok a cikkekhez tartozó kulcsszavakat, ill. a CIKK táblához kapcsolatot biztosító elsődleges kulcsokat tárolják idegen kulcsként (CikkCím). A séma részletesen:

- CIKK (CikkCím, SzekcióNév, OldalKezd, OldalVégső, Nyelv, ForrásSzám)
- KULCSSZÓMAGYAR (CikkCím, KulcsszóMagyar)
- KULCSSZÓANGOL (CikkCím, KulcsszóAngol)

Az ER-modell a Cikk egyedtípusban is több leszármaztatott attribútumot tartalmaz, amelyeket a relációs modell táblázatai nem fognak tartalmazni. Ezek a tulajdonságok a következők: NSzerzőDb, IdegenCikkDb, SzekcióDb, NemMkCikkDb, MkCikkDb, CikkSzámDb, CikkOldalSzámDb.

4. Kapcsolatok: Az ER-modell alapján minden kapcsolat totális. A FOLYÓIRAT és a SZERZŐ között, valamint a SZERZŐ és a CIKK között több–a–többhöz (N:M) kapcsolat van; a kapcsolatot biztosító táblázat neve PUBLIKÁL, ill. ÍR. A CIKK és a FOLYÓIRAT között egy–a–többhöz (1:N) kapcsolatot kell létrehozni, a táblázat neve TARTALMAZ. A kapcsolat táblázatok sémája:
  - PUBLIKÁL (Évfolyam, Szám, SzerzőAzonosító)
  - ÍR (SzerzőAzonosító, CikkCím)
  - TARTALMAZ (Évfolyam, Szám, CikkCím)

#### 4. Záró gondolatok

A létrehozott relációs-modell alkalmas a megvalósításra, de célszerű ellenőrizni – akár tesztadatokkal –, hogy (1) nem tartalmaz-e anomáliákat, (2) lehet-e, érdemes-e tovább csökkenteni az adatokban lévő redundanciát, ill. (3) megfelel-e legalább a 3. normál formának (3NF) és ezek függvényében akár további finomítás, módosítás is szükséges lehet.

A létrehozott relációs modell ellenőrzése után következhet az adatbázis-tervezés következő fázisa, a fizikai megvalósítás. Ez a relációs séma SQL-alapú megvalósítását jelenti a kiválasztott relációs adatbázis-kezelő rendszerben.

#### Irodalomjegyzék

- Badia, A., Lemire D. (2011): A Call to Arms: Revisiting Database Design. *Sigmod Record*, 40 (3): 61–69. <https://doi.org/10.1145/2070736.2070750>
- Dey, D., Storey, V., Terence, B. (1999): Improving Database Design Through the Analysis of Relationships. *ACM Transactions on Database Systems*, 24 (4):. 453–486. <https://doi.org/10.1145/331983.331984>
- Fabulya Z. (2018a): Access alkalmazás kialakítása ügyfélközpontú szolgáltatások nyilvántartására. *Jelenkori Társadalmi és Gazdasági Folyamatok*, 13 (1-2): 67–76.
- Fabulya Z. (2018b): Access alkalmazás kialakítása dolgozói jelenlét nyilvántartására. *Jelenkori Társadalmi és Gazdasági Folyamatok*. 13 (1-2): 151–160.
- Halassy B. (2000): *Adatmodellezés. Elmélet és gyakorlat*. Budapest. <<https://mek.oszk.hu/11100/11144>> (2020. 08.10.)
- Hampel Gy., Heves Cs. (2019): *Informatika alapjai mérnököknek, alapszakos hallgatók számára*. Szegedi Tudományegyetem, Szeged.

- Kovács L. (2004): *Adatbázisok tervezésének és kezelésének módszertana*. Computer Books, Budapest.
- Kroenke, D. M., Gray, C. D. (2006): Toward a Next Generation Data Modeling Facility: Neither the Entity-Relationship Model nor UML Meet the Need. *Journal of Information Systems Education*, 17 (1): 29–38.
- Silberschatz, A., Korth, H. F., Sudarshan, S. (1996): Data Models. *ACM Computing Surveys*, 28 (1): 105–108. <https://doi.org/10.1145/234313.234360>
- Silberschatz, A., Korth, H. F., Sudarshan, S. (2020): *Database System Concepts*. 7th edition. McGraw-Hill Education, New York.
- Szabó B. (2013): *Adatbázis fejlesztés és üzemeltetés I*. Eszterházy Károly Főiskola, Eger.
- Tímár L., Vigh K., Tátrai J., Szigeti J., Vathy Á., Telekesi É., Vass I., Kocsis T., Priskinné R. Zs., Erdélyiné M. (19997): *Építsünk könnyen és lassan adatmodellt!* Veszprémi Egyetem & Műszertechnika-Veszprém Kft., Veszprém.
- Ullman, J., Widom, J. (2008): *Adatbázisrendszerek – Alapvetés*. Panem Kiadó, Budapest.