

## PREPARING THE CONCEPTUAL MODEL OF A DATABASE

György HAMPEL

***Abstract:** Since 2006, several Hungarian and foreign experts have written publications in the Journal of Contemporary Social and Economic Processes. To efficiently extract the information from the publications, the idea to create a database arose. The first step to achieve this is to prepare a conceptual model which is the subject of the current article. Systematic reflection on the task – what to include as an entity set, what attributes should be used, what kind of relationships are needed – helped to create a conceptual model which is suitable for implementation, and it may give new ideas to the editorial board to extract additional information, which may require further additions and modifications before creating the final version of the database. The created entity-relationship model includes 3 entity sets, 39 attributes and 3 relationships.*

***Keywords:** database, data model, entity-relationship model, journal*

### 1. INTRODUCTION

The first yearbook of the Institute of Economics and Rural Development of the University of Szeged Faculty of Engineering was published in 2006. The book contained peer-reviewed scientific publications of the professors and lecturers working at the institute. The authors presented a small piece of their research topics. In 2007, a new scientific yearbook was published with the title “Contemporary social and economic processes” with articles by external lecturers as well. While keeping the title, the book was converted to a journal and from 2008 onwards, the range of authors was further expanded with Hungarian and foreign authors from different institutes. Since then, a total of 24 volumes were published. Over the years, more than 450 articles were written by nearly 500 different authors from a wide range of disciplines. Seeing the unbroken development, the idea arose at the Institute of Engineering Management and Economics taking care of the journal, that a journal database should be created that could make it easier to extract relevant information from the articles published in the issues.

So, the task is to create a database. To do this, first, we need to design the database using a high-level conceptual model. A very simple tool, the entity-relationship model is perfectly suitable for this purpose. Then the conceptual model has to be converted to a logical model. Considering the type of data, the relational model is a perfect solution. The last step is to physically create the database with a relational database management system that uses SQL (Structured Query Language).

In this publication, I deal with the conceptual design of the database related to the journal, i. e. the preparation of the entity-relationship model. The benefits of a thoughtful design are:

- It displays the structure of the database in an easily comprehensible form with the data groups and their relationships, as well as any restrictions.
- As the model is easy to understand, it facilitates the dialogue between the user and the programmer.
- It gives new ideas to both users and programmers.
- It is a plan or description that does not need to be changed by selecting or modifying the database management system.

## 2. DATA AND INFORMATION

We usually treat data and information as synonyms in everyday life, while in many areas of science, specialists separate the two concepts. There are many similar interpretations if we look up dictionaries. About data we can read, for example:

- According to Meriam-Webster Dictionary, data is: “factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation”, “...information in digital form that can be transmitted or processed”, “... information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful”. (Meriam-Webster1, 2021)
- In Cambridge Dictionary: “Information, especially facts or numbers, collected to be examined and considered and used to help decision-making, or information in an electronic form that can be stored and used by a computer.” (Cambridge1, 2021)
- On YourDictionary website: “Data is defined as facts or figures, or information that's stored in or used by a computer.” “Facts that can be analyzed or used in an effort to gain knowledge or make

decisions; information.” “Statistics or other information represented in a form suitable for processing by computer.” “Facts or figures to be processed; evidence, records, statistics, etc. from which conclusions can be inferred; information.” “Information in a form suitable for storing and processing by a computer.” (Yourdictionary1, 2021)

- In the Glossary of Statistical Terms of OECD: “Data is the physical representation of information in a manner suitable for communication, interpretation, or processing by human beings or by automatic means.” (OECD1, 2021)

Some of the many information concepts, in the same dictionaries:

- Meriam-Webster Dictionary: Information is “knowledge obtained from investigation, study, or instruction”, “a signal or character (as in a communication system or computer) representing data”, “something (such as a message, experimental data, or a picture) which justifies change in a construct (such as a plan or theory) that represents physical or mental experience or another construct”, “the communication or reception of knowledge or intelligence.” (Meriam-Webster2, 2021)
- Cambridge Dictionary: “facts about a situation, person, event, etc.”, “news, facts, or knowledge”. (Cambridge2, 2021)
- YourDictionary: “Knowledge or facts learned, especially about a certain subject or event.”, ... news or knowledge received or given.” (Yourdictionary2, 2021)
- Glossary of Statistical Terms of OECD: “Information (in information processing) refers to knowledge concerning objects, such as facts, events, things, processes, or ideas, including concepts, that within a certain context has a particular meaning.” (OECD2, 2021)

These are only a few examples, how data and information can be defined. But as mentioned earlier, we usually do not distinguish between the two concepts. To get more out of the data (make more data or information from the existing ones), data has to be collected, stored, processed and communicated to the user.

Data (and information) can be effectively stored in a properly designed database. There are several benefits to creating and using a database, such as (Kroenke 2006, Kacsukné–Kiss 2009):

- a uniform, logically clear data structure,
- the data and the programs that manage them are independent of each other so that either can be modified without modifying the other,

- the data can be queried flexibly.

Databases can be created and managed without computers as well, but not using computers for this kind of task is not efficient anymore.

Just as in the case of data and information, several of definitions can be enumerated, as there is no uniformly accepted definition, for example:

- A database consists of related data and also metadata describing data types and relationships; it also includes a data management system (Tímár et al. 1997).
- The database is a set of a finite number of entity occurrences, their finite number of property values and relationship occurrences, organized according to the data model. (Halassy 1994).
- The database is an integrated data structure that stores data about several different objects in an organized and persistent manner according to a data model with auxiliary information (so-called meta-data) to ensure efficiency, integrity and protection (Kovács 2004).
- According to Oracle Inc. “A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.” (Oracle, 2021)

### 3. DESIGNING A DATABASE

When designing the database, we first create a conceptual model, then a logical model is created based on that, and only after that should the actual, physical implementation commence.

A conceptual data model is a collection of tools designed to describe reality in such a way that the created model can answer questions about reality (Watson 2006). This model also does not address data structure or physical storage problems.

The creation of a conceptual (high-level) data model plays a significant role in the database design process, both in theory and in practice (Dey et al. 1999). Conceptual schemas include the set of entities, their characteristics, relationships, and constraints. Also, the logical structure of the database is illustrated with diagrams; they illustrate reality in a way that is comprehensible to a person unfamiliar with developing databases (Tímár et al. 1997). Of course, like everything, this can go in the wrong direction

as well: Improper use of modelling tools, limitations of the imagination of users and database designers, their different visions and approaches to problems contribute to poor design (Badia–Lemiere 2011) and, although the model toolbox is simple, it is essential that the modeller learns or has the ability to correctly identify the necessary entities (objects) and their relationships (Watson 2006) under the given circumstances.

According to Watson (2006), those who are not yet proficient enough in data modelling (e.g., university students studying database design) make common mistakes like:

- they do not recognize that something thought to be an attribute is, in fact, an entity,
- they are unable to generalize, merge entities and assign them to entity sets,
- relationships are not checked from both directions, so the determination of cardinality (the number of related entities) is incorrect, and
- they ignore possible exceptions.

Some useful principles to consider when designing a database are: (Ullman–Widom 2009)

- Use realistic modelling: Entity sets, attributes, and relationships must reflect reality (for example, ARTICLE entity cannot have a scientific degree attribute).
- Try to be free from data redundancy: Every effort should be made to include all data only once. For example, do not store the author's name as an attribute in the ARTICLE and then again in the AUTHOR entity set. This avoids problems called (update, deletion, and insertion) anomalies that make it difficult to use and maintain the database (Halassy 2000). Although, there may be cases when, for some reasons, we can still be forced to store redundant data; this depends on the given circumstances, requirements and the database management system.
- Make it simple: Do not add more entity sets, attributes, and relationships than are absolutely necessary. For example, in the case of AUTHOR, there is no point in including the eye colour attribute.
- Choose the right relationships: Entity sets – especially if there are many of them – can be connected in several ways. It is usually not advisable to create all the relationships, as this can lead to the anomalies mentioned earlier and only make it difficult (or even impossible) to manage the database. It is advisable to mark the relationship(s) that are necessary for the current problem and to do

this, we need to know in advance what we actually expect from the database. For example, it is advisable to create many-to-many connections between the AUTHOR and the ARTICLE, as an author may write several articles and several articles may have several co-authors, but at the same time, there may be only one-to-many links between the JOURNAL and the ARTICLE, since an article can normally be published only once in a journal and, on the other hand, a journal issue can contain several articles.

- Choose the appropriate type of objects: There are several ways to describe reality. We may be able to describe a slice of reality with an entity set, an attribute or a relationship. In general, attributes are easier to incorporate into the model than an entity set or relationship. However, solving everything with attributes is not always practical, because it makes it difficult to manage the final database (see Fig. 1.).

#### 4. THE ENTITY-RELATIONSHIP MODEL

In 1976, a 29-year-old computer scientist, Peter Pin-Shan Chen laid the foundations of the Entity-Relationship model (ER-model) with his article (Chen 1976). This publication had a significant and long-lasting effect on data modelling.

The ER-model is considered to be the most popular tool for conceptual data modelling (Hartmann, 2003); In addition to its simplicity, its popularity is due to its theoretical validity and its excellent suitability for the preparation of a relational data model used by many of today's database management systems. As a result, the ER-model is taught in many higher education institutions, which also contributes to its popularity and prevalence (Carte et al. 2006).

In the model, the so-called Entity-Relationship Diagram (ERD) is used to represent the structure of the data (Ullman–Widom 2009). The ERD is a graph where there are three basic elements: (1) the entity set(s) (denoted by rectangles), (2) the attribute(s) (written in ellipses) and (3) the relationships (rhombuses) (Silberschatz et al. 1996, Tímár et al. 1997, Szabó 2013). Entity sets are connected to their attributes and relationships using lines, which are the edges of the graph (Ullman–Widom 2009).

Thus, the ER-model is expressive, simple and even non-professionals can understand it relatively easily. This model uses only a few concepts, so it can be learned quickly. It works with illustrative diagrams, but due to the representation method used, very complex databases can be difficult to describe, and the resulting diagram may be difficult to

understand (Dunn et al. 2005). The basic elements of the ER-model (Tímár et al. 1997, Ullman–Widom 2009):

1. An entity is an abstract object that can be named with a noun. Similar entities form a set of entities, for example, JOURNAL, ARTICLE, AUTHOR.
2. The attribute belongs to the entity set or relationship. It describes its property or properties (for example: in the case of an author: name, academic degree, job, gender). An entity set must have at least one attribute, while it is not necessary to assign attribute(s) to a relationship. Nouns are commonly used when naming attributes and the types are:
  - simple (ellipse with a single line),
  - composite (can be broken down into several sub-attributes, an ellipse drawn with a single line),
  - multivalued (ellipse with a double line), and
  - derived (ellipse with a dashed line; can be calculated from the values of other properties, but not indicated how) (Hampel–Heves 2019).
3. A relationship connects two – or even more – sets of entities and a verb is used to describe that. The connecting is binary when two entity sets are connected, and this is the most common, but the model allows connecting any number of entity sets (Ullman–Widom 2009).

According to a study by Dey et al. (1999), the interpretation of entity sets and their attributes does not cause much problem for the users of the model, however, this cannot be said for the interpretation of relationships.

The types of relationships can be (Tímár et al. 1997, Dey et al. 1999, Ullman–Widom 2009):

- recursive (self-referential),
- 1:1 (one-to-one),
- 1:N (one-to-many),
- N:M (many-to-many),
- or in the case of non-binary relationships, depending on the number of sets of individuals, we can speak of n-degree relationships (n: number of entity sets in the relationship).

In addition, all listed relationships can be total and partial (optional). A relationship is total when all entities are involved in a relationship within an entity set and is partial when there is at least one entity that has no relationship to any entities to the other entity set (Dullea et al. 2003).

The key attribute plays an important role in this model which is indicated on the ERD by underlining the attribute name. A key is an attribute that is capable of uniquely identifying an entity. If there is more than one attribute that is suitable for the purpose, the one we select is the primary key and the others will be alternative keys. It may not be possible to identify an individual with one attribute alone, but by combining the minimum number of required attributes to obtain a key suitable for unambiguous identification of an entity (Ullman–Widom 2009).

There are enhanced versions of the ER-model described above (Enhanced ER model, EER model, extended ER-model) which add additional possibilities to the model's ability to describe reality, such as introducing major and subclasses, generalization and specialization. concepts (Tímár et al. 1997, Ullman–Widom 2009), but in our case, these do not play a significant role in the database to be created.

It should also be noted that several variants of the representation of ER-diagrams coexist (Hartmann 2003), which differ mainly in the representation of relationships and their nature (Chua–Storey 2011), as can be seen in Table 1. Although, it is not the mode of representation that primarily matters, but the ability to model correctly (Watson 2006).

Table 1. Two data modelling “dialects”

Relationship	Minimalist	DB Visual Architect
1:1		
1:1 with weak entity type		
1:N		
1:N with weak entity type 1		

Source: Author’s edit based on Watson (2006)

### 5. THE COMPLETED ER-MODEL

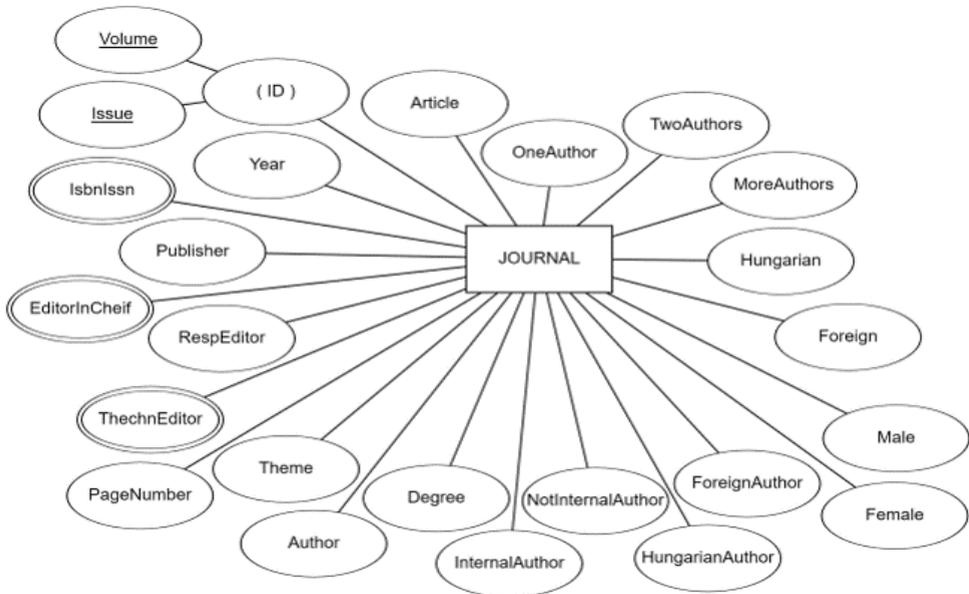
After collecting and overviewing the requirements, an ER-model was created, which summarizes the information that should be extracted from the database in a single entity type and its associated attributes (fig. 1.).

It can be seen that this ER model, consisting of a single entity set with over twenty attributes, is capable of extracting a lot of information, but requires a lot of preparation, preliminary manual aggregation, and does not contain important basic data (such as authors' names, article titles etc.):

1. ID (composite attribute): identifies each issue.
2. Volume (simple attribute, part of ID): journal volume.

3. Issue (simple attribute, part of ID): issue within volume  
Volume and Issue together is the primary key.
4. Year (simple attribute): publishing year of issue.
5. IsbnIssn (multivalued): ISBN or ISSN numbers of the journal.
6. Publisher: (simple attribute): name (and other data) of the publisher; can be composite attribute as well.
7. EditorInChief (multivalued): name (and other data) of the editor-in-chiefs.
8. RespEditor (simple attribute): the name (and other data) of the editor responsible for the given issue.
9. TechnEditor (multivalued): name (and other data) of persons responsible for editing the issue.
10. PageNumber (simple attribute): total page number of the issue.
11. Theme (simple attribute): total number of themes in the issue.
12. Article (simple attribute): total number of scientific articles in the issue.
13. OneAuthor (simple attribute): total number of articles with a single author.
14. TwoAuthors (simple attribute): total number of articles with two authors.
15. Moreauthors (simple attribute): total number of articles with more than two authors.
16. Hungarian (simple attribute): total number of Hungarian language articles in the issue.
17. Foreign (simple attribute): total number of non-Hungarian language articles in the issue.
18. Author (simple attribute): total number of authors in each issue.
19. InternalAuthor (simple attribute): number of authors of our Faculty in an issue.
20. NotInternalAuthor (simple attribute): number of authors not from our Faculty in an issue.
21. Degree (simple attribute): number of authors with degrees in an issue.
22. HungarianAuthor (simple attribute): number of Hungarian authors in each issue.
23. ForeignAuthor (simple attribute): number of not Hungarian authors in each issue.
24. Male (simple attribute): number of male authors in each issue.
25. Female (simple attribute): total number of female authors in each issue.

Figure 1. The ER-diagram of the journal (version 1)



Source: Author's edit with erdplus.com data modelling website

After reviewing the first version of the ER-model, a second, final one was created. This model contains three entity sets (JOURNAL, ARTICLE, AUTHOR) with attributes, as well as three relationships (writes, publishes, contains). The diagram of the completed ER-model is shown in Fig. 2. The description of this model is as follows:

Entity set JOURNAL:

1. Volume (simple attribute): journal volume.
2. Issue (simple attribute): number of issue within volume.  
Volume and Issue together is the primary key.
3. Year (simple attribute): publishing year of issue.
4. IsbnIssn (multivalued): ISBN or ISSN numbers of the journal.
5. Publisher: (simple attribute): name (and other data) of the publisher; can be composite attribute as well.
6. EditorInChief (multivalued): name (and other data) of the editor-in-chiefs.
7. RespEditor (single attribute): the name (and other data) of the editor responsible for the given issue.
8. TechnEditor (multivalued): name (and other data) of persons responsible for editing the issue.
9. PageNumber (simple attribute): total page number of the issue.

10. TotalPages (derived, calculated attribute): total page number of all the issues.

Entity set ARTICLE:

1. ArticleTitle (simple attribute and primary key): there can be no two or more articles with the same title.
2. SectionTitle (simple attribute): The section title of the theme or section (if any) where the article belongs.
3. PageStart (simple attribute): first page number of the article.
4. PageEnd (simple attribute): last page number of the article.
5. Language (simple attribute): the language of the article.
6. KeywordHun (composite): Hungarian keywords of the article.
7. KeywordEng (composite): English keywords of the article.
8. ReferencesNum (simple attribute): number of referenced sources in the article.
9. TotalPages (derived attribute): total page number of the article.
10. TotalArticles (derived attribute): total number of articles.
11. SectionSum (derived attribute): total number of sections.
12. NAuthorSum (derived attribute): number of articles written by N authors.
13. HunArticleSum (derived attribute): number of articles written by Hungarian authors.
14. ForeignArticleSum (derived attribute): number of articles written by foreign authors.
15. FacultyArticleSum (derived attribute): number of articles written by colleagues at our faculty,
16. NFacultyArticleSum (derived attribute): number of articles written by authors not working at our faculty.

Entity set AUTHOR:

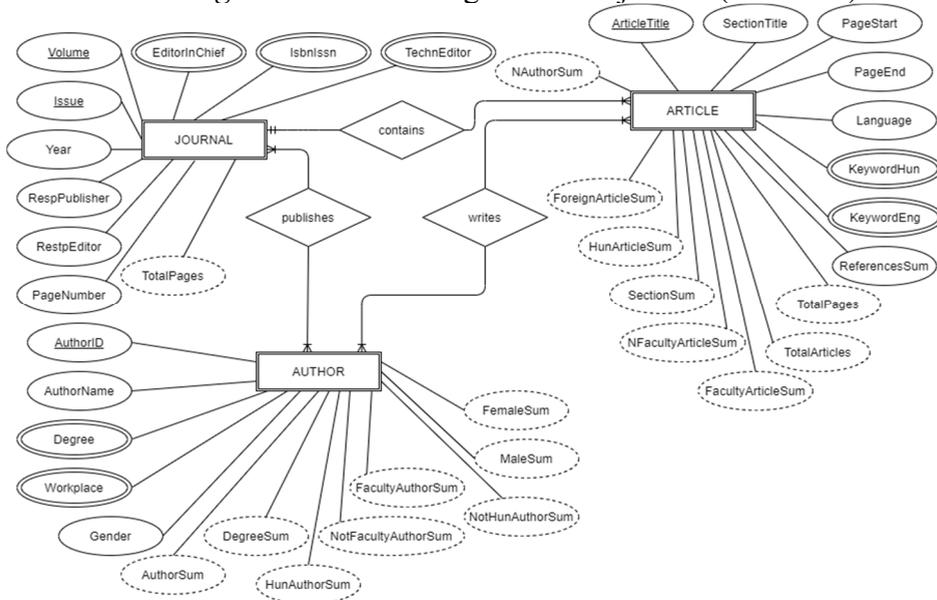
Careful consideration should be given to the attribute(s) by which each author can be identified. The name alone is not enough, as two or more different authors may have the same name. If someone changes his or her name between two publications, that will mean two names and will therefore count as two different people. Nor do we get any further by merging the name, the degree (and job) attributes: if an author who publishes multiple times gets a (new) academic degree or changes jobs, he will again count as two different people. The problem can be handled with an arbitrarily created unique ID that can be used as a primary key.

1. AuthorID (simple attribute): primary key to identify each author.
2. AuthorName (simple attribute): name of the author.

3. Degree (multivalued attribute): degree of the author when sending the finalized version of the article; an author can have more than one scientific degrees.
  4. Workplace (multivalued attribute): data of the author's workplace(s).
  5. Gender (simple attribute): gender of the author (based on the name).
  6. AuthorSum (derived attribute): number of authors of an article.
  7. DegreeSum (derived attribute): number of authors with degree in an article.
  8. HunAuthorSum (derived attribute): number of Hungarian authors in an article.
  9. NotHunAuthorSum (derived attribute): number of foreign authors in an article.
  10. FacultyAuthorSum (derived attribute): number of authors of an article working at our Faculty.
  11. NotFacultyAuthorSum (derived attribute): number of authors of an article not working at the Faculty.
  12. FemaleSum (derived attribute): number of female authors of an article.
  13. MaleSum (derived attribute): number of male authors of an article.
- The relationships between the entity sets:
- 1 issue can contain articles from many authors, 1 author can publish in many issues, 1 article can only be published in 1 issue.
  - Each author must publish in one of the issues: an author must have at least 1 relationship to 1 article and can be author of many articles; an author must have at least 1 relationship to a journal issue and can have many relationships to journal issues.
  - All articles must be published in one of the issues: an article can have 1 and only 1 relationship to the journal; an article must have at least 1 relationship with 1 author and can have relationship to multiple authors.
  - All journal issues must have an author: a journal issue must have at least 1 relationship with 1 author and can have multiple authors.
  - Each journal issue must have an article: a journal issue must have at least 1 relationship to an article.
  - As a result of the above, all relationships are total. A many-to-many (N:M) relationship must be established between the JOURNAL and the AUTHOR, and between the AUTHOR and the ARTICLE, and

a one-to-many (1:N) relationship must be created between the ARTICLE and the JOURNAL.

Figure 2. The ER-diagram of the journal (version 2)



Source: Author’s edit with tools provided by diagrams.net website

## 6. CONCLUSIONS

Systematic reflection on what seemed first to be a task that is simple and quick to implement – what to include as an entity set, what attributes should be used, what kind of relationships are needed – helped to create a correct conceptual model.

The created ER-model in its current form is suitable for implementation, and it may be suitable to give new ideas to the editorial board to extract additional information, which may require further additions and modifications before creating the final version of the database.

The next phase of the work is to transfer the conceptual model to a logical level, i. e. to create a relational model. In this step, we need to convert the entity sets, attributes, and relationships to a relational schema that describes tables and their structure, as well as the relationship between the tables.

The final phase will be the physical implementation, which is the creation of the SQL-based version of the relational schema in the selected relational database management system.

In case of further needs, this database can be developed to be suitable for text mining, and in this case, it can contribute even more effectively to the utilization of the knowledge inherent in the journal “Contemporary Social and Economic Processes”.

## References

- Badia, A., and Lemire, D. A. 2011. “Call to Arms: Revisiting Database Design.” *Sigmod Record* 40 (3): 61–69.
- Cambridge1. “Cambridge Dictionary: Data.” Cambridge University Press. Last accessed March 25, 2021. <https://dictionary.cambridge.org/dictionary/english/data>
- Cambridge2. “Cambridge Dictionary: Information.” Cambridge University Press. Last accessed March 25, 2021. <https://dictionary.cambridge.org/dictionary/english/information>.
- Carte, T. A., and Jaspersen, J., Cornelius, M. E. 2006. “Integrating ERD and UML Concepts When Teaching Data Modeling.” *Journal of Information Systems Education* 17 (1): 55–63.
- Chen, P. P.-Sh. 1972. “The Entity-Relationship Model – Toward a Unified View of Data.” *ACM Transactions on Database Systems* 1 (1): 9–36.
- Chua, C. E. H., and Storey, V. C. 2011. “Issues and Guidelines in Modeling Decomposition of Minimum Participation in Entity-Relationship Diagrams.” *Communications of the Association for Information Systems* 29 (9): 159–184.
- Dey, D., and Storey, V., Terence, B. 1999. “Improving Database Design Through the Analysis of Relationships.” *ACM Transactions on Database Systems* 24 (4): 453–486.
- Dullea, J., and Song, Il-Y., Lamprou, I. 2003. “An Analysis of Structural Validity in Entity-Relationship modeling.” *Data & Knowledge Engineering* 47 (2): 167–205.
- Dunn, Ch. L., and Gerard, G. J., Grabski, S.V. 2005. “Critical Evaluation of Conceptual Data Models.” *International Journal of Accounting Information Systems* 6 (2): 83–106.
- Halassy, B. 1994. *Az adatbázis-tervezés alapjai és titkai [Basics and secrets of database design]*. Budapest: IDG Magyarországi Lapkiadó Kft. Accessed March 25, 2021. <https://mek.oszk.hu/11100/11123/11123.pdf>
- Halassy, B. 2000. *Adatmodellezés. Elmélet és gyakorlat [Data modeling. Theory and practice]*. Budapest: Hallassy Béla. Accessed March 25, 2021. <https://mek.oszk.hu/11100/11144/11144.pdf>
- Hampel, Gy., and Heves, Cs. 2019. *Informatika alapjai mérnököknek, alapszakos hallgatók számára [Basics of IT for engineers, undergraduate students]*. Szeged: Szegedi Tudományegyetem.
- Hartmann, S. 2003. Reasoning about participation constraints and Chen's constraints. *Database Technologies 2003, Proceedings of the 14th Australasian Database Conference*, 2003: 105–113. Adelaide, South Australia: ADC.
- Kacsukné, B. L., and Kiss, T. 2009. *Bevezetés az üzleti informatikába [Introduction to information science]*. Budapest: Akadémiai Kiadó.
- Kovács, L. 2004. *Adatbázisok tervezésének és kezelésének módszertana [Database design and management methodology]*. Budapest: Computer Books.

- Kroenke, D. M. 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.
- Meriam-Webster1. "Meriam-Webster Dictionary: Data." Merriam-Webster Inc. Last accessed March 25, 2021.  
<https://www.merriam-webster.com/dictionary/data#synonyms>
- Meriam-Webster2. "Meriam-Webster Dictionary: Information." Merriam-Webster Inc. Last accessed March 25, 2021.  
<https://www.merriam-webster.com/dictionary/information>
- OECD1. "Glossary of Statistical Terms: Data." OECD. 2021. Last accessed March 25, 2021. <https://stats.oecd.org/glossary/detail.asp?ID=532>
- OECD2. "Glossary of Statistical Terms: Information." OECD. 2021. Last accessed March 25, 2021. <https://stats.oecd.org/glossary/detail.asp?ID=4900>
- Oracle. "Database/What Is a Database?" Oracle Corp. Last accessed March 25, 2021.  
<https://www.oracle.com/database/what-is-database/>
- Silberschatz, A., and Korth, H. F., Sudarshan, S. 1996. "Data Models." *ACM Computing Surveys* 28 (1): 105–108.
- Szabó, B. 2013. *Adatbázis fejlesztés és üzemeltetés I [Database development and operation I]*. Eger: Eszterházy Károly College.
- Tímár, L., and Vigh, K., Tátrai, J., Szigeti, J., Vathy, Á., Telekesi, É., Vass, I., Kocsis, T., Priskinné, R. Zs., Erdélyiné, M. 1997. *Építsünk könnyen és lassan adatmodellt! [Let's build a data model easily and slowly!]*. Veszprém: University of Veszprém & Műszertechnika-Veszprém Kft.
- Ullman, J., and Widom, J. 2009. *Adatbázisrendszerek – Alapvetés [Database systems – Basics]*. Budapest: Panem Kiadó.
- Watson, R. T. 2006. "The Essential Skills of Data Modeling." *Journal of Information Systems Education* 17 (1): 39–41.
- Yourdictionary1. "Yourdictionary: Data." LoveToKnow. Last accessed March 25, 2021.  
<https://www.yourdictionary.com/data>
- Yourdictionary2. "Yourdictionary: Information." LoveToKnow. Last accessed March 25, 2021. <https://www.yourdictionary.com/information>

#### NOTES ON THE AUTHORS

**György HAMPEL** (Dr. Ph.D.) Economists, associate professor. University of Szeged, Faculty of Engineering Institute of Engineering Management and Economics. Address: Mars ter 7. H-6724 Szeged, Hungary. Phone number: +36 62 546 000. Email address: [hampel@mk.u-szeged.hu](mailto:hampel@mk.u-szeged.hu). Main research areas: information management, e-learning. Main courses: information management, information systems, information technology, economic statistics, database design and database management, data and information visualization, project management.