When Entity/Relationship Models Meet Graph Databases

Philipp Skavantzos
The University of Auckland
Auckland, New Zealand
philipp.skavantzos@auckland.ac.nz

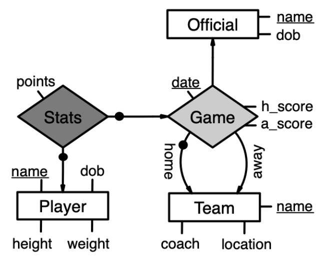
Sebastian Link
The University of Auckland
Auckland, New Zealand
s.link@auckland.ac.nz

ABSTRACT

This tutorial shows how traditional Entity/Relationship modeling and modern graph data modeling can be combined to bring forward well-designed graph data models that process workloads and maintain data integrity efficiently.

PVLDB Reference Format:
Philipp Skavantzos and Sebastian Link. When Entity/Relationship Models Meet Graph Databases. PVLDB, 18(12): 5444 - 5447, 2025.
doi:10.14778/3750601.3750617

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/lshelloworld/Tutorial-GraphModelling.



(a) E/R design

- TEAM={*name*, *coach*, *location*}
- OFFICIAL={*name*, *dob*}
- PLAYER={*name*, *dob*, *height*, *weight*}
- GAME={*hname*, *aname*, *oname*, *date*, *h_score*, *a_score*} with foreign keys
  - [*hname*] ⊆ TEAM[*name*]
  - [*aname*] ⊆ TEAM[*name*]
  - [*oname*] ⊆ OFFICIAL[*name*]
- STATS={*pname*, *hname*, *date*, *points*} with foreign keys
  - [*pname*] ⊆ PLAYER[*name*]
  - [*hname*, *date*] ⊆ GAME[*hname*, *date*]

(b) Matching relation schemata

Figure 1: E/R diagram and relational database schema

1 MOTIVATION

Graph databases offer exciting opportunities for advancing data management [18]. The broad and deep landscape of approaches to managing graph data will soon converge towards a standard, most likely the property graph model [7]. Despite the popularity and growing maturity of graph database systems[1], they continue to rank well below trusted relational technology. We further uptake of graph databases we need a principled methodology for their design, rigorous schema and data integrity support. As a consequence, academics and practitioners have worked together to develop proposals for standards of query [1, 2], schema [3], and data integrity [4] languages. A common theme is flexibility and expressiveness to support dynamic and open environments for complex applications. However, comprehensiveness may be a call for concern not to commit mistakes similar to those made for XML, which became so expressive [5, 16] that people stopped using it and replacement was found in JSON which is not data-centric at all. In fact, simplicity, familiarity and maturity are cornerstones for making systems accessible to a broad range of people that can use it confidently. For the emerging standards PG-Schema [3] for graph schemata and PG-Key [4] for graph integrity it is not yet well-understood which of their fragments support which applications. PG-Schema, in particular, supports basic features of Entity/Relationship (E/R) models, but its expressiveness is well beyond those capabilities.

Indeed, Chen's E/R model [8] constitutes a best breed of conceptual data models. The model captures entities and their relationships in an easy-to-understand framework powerful enough to derive a formal data model. E/R models represent complex requirements for the target database visually [9]. Indeed, the graphical depiction of an E/R diagram is invaluable for effective communication between experts with different expertise. Fig. 1a depicts an E/R diagram for a basketball app. Entity types are visualized as rectangles with their attributes, relationship types are visualized as diamonds with their attributes and directed edges (called E/R links), pointing to each of its components, that is, those object types (entity or other relationship types) for which they define a relationship. The E/R key of an object type consists of its attributes that are underlined and its components with a dot on their E/R link. For example, the E/R key of GAME is {{*home*:TEAM},{*date*}}. Fig. 1b shows matching relation schemata, with attributes of the key underlined.
E/R modeling is a methodology for generating E/R diagrams that are well-designed as they guarantee data integrity and do not exhibit data redundancy or update anomalies in any instances [14, 15, 20].

Given these developments, the tutorial aims at addressing the following ambitious research question.

> What methodology for designing property graph schemata can process workloads and maintain data integrity efficaciously?

In what follows, Sec. 2 will describe the target audience and the main points they will take away from the tutorial. These takeaways are fundamental for conceptual, graph and logical data modeling. The scope and depth of the tutorial will be outlined in Sec. 3. Details about the length, style, presenters and previous offerings of the tutorial will be given in Sec. 4.

2 TAKEAWAYS

We will name the primary interest group for this tutorial, but also the general audience able to understand the main takeaways presented in the tutorial. Next we will describe in high level terms what these main takeaways are.

Target audience. The specific target audience are people with an interest in either graph or conceptual data modeling, or database

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750617

[1] https://db-engines.com/en/ranking_trend

design. In fact, the tutorial will bring these areas together and show their positive impact on one another. The topic is timely as graph databases have grown large in popularity among researchers and practitioners, but also because the design of graph databases has emerged as a new and important direction.

However, the tutorial is accessible for computer science students at undergraduate level. Background material will be motivated and introduced from scratch, so even students with little knowledge on databases will still understand the main takeaways. In fact, the tutorial style described later will ease understanding of the subject.

What will they learn? We will showcase the fundamental impact E/R modeling and graph databases have on one another.

Firstly, property graphs that comply with E/R diagrams, called E/R graphs, constitute the first kind of E/R databases that are graphs. Hence, instead of translating E/R diagrams to other data models, such as relational or web models, we can directly manage property graphs that comply with E/R diagrams. As immediate consequence, native graph database systems can manage E/R databases.

E/R modeling is the most well-known methodology to bring forward databases that are well-designed, formally achieved by being: acyclic, key-based, and foreign-key based. Indeed, they provide a foundation for eliminating sources of inconsistency and data redundancy, processing updates consistently and efficiently, specifying queries soundly and evaluating them quickly. *As the second fundamental point, we show that E/R diagrams form a core fragment of PG-Schema that captures well-designed (graph) databases.* This establishes PG-Schema as a general data modeling tool as well.

Thirdly, the audience will learn about three principled approaches to managing entity and referential integrity efficiently in E/R graphs. Interestingly, directed edges in property graphs take on the exclusive role of maintaining referential integrity, while property-value pairs are reserved to reside on nodes only. These insights facilitate efficient i) udpates as sources of data redundancy and inconsistency are eliminated, and ii) join queries as key/foreign key joins are directly represented as directed edges between nodes. Furthermore, by linking nodes from different graphs, data integration becomes effective. Because edges cannot be targets of directed edges, it is impossible to reference data residing on edges.

Fourthly, we demonstrate that reasoning about PG-Key is infeasible, but identify E/R keys as a fragment of PG-Key that can help efficiently manage data integrity in well-designed property graphs.

The audience will learn further (i) how E/R diagrams and property graphs can be combined to unify conceptual, logical and graph data modeling, (ii) how integrity management is taken to the next level by eliminating property redundancy, and (iii) why relational benchmarks perform well when translated into property graphs.

Indeed, the tutorial will show how a major inhibitor to the uptake of graph databases can be turned into a strong driver.

3 SCOPE AND DEPTH

We will give an overview of the scope and depth of what the tutorial will cover. The following subsections form the logical units of the tutorial in the sequence they will be presented in.

Overview: Motivation and Goals. The tutorial will start by showing how popular graph database systems have become recently, but also by contrasting their popularity to that of relational technology. This serves as a motivation for asking why the audience believes that is. One key reason is the lack of a methodology enabling people to design graph databases well enough to ensure efficacious processing of their target workload and maintaining data integrity. This will lead to the research question, after which the scope and goals of the tutorial will be outlined. Query languages for graph databases are out of scope, with comprehensive surveys already available [1, 2, 7].

Background on E/R modeling. We will illustrate important concepts from E/R modeling on our basketball application. This includes schemata and diagrams, their instances, E/R keys that manage entity integrity, and E/R links that manage referential integrity. Particular emphasis will be placed on the ability to express relationship types of arbitrary arity and order.

Using our running example application, we will discuss the Entity/Relationship model [20] as a methodology for designing databases well. The discussion will bring forward three central pieces that formalize what the notion of *well-designed* shapes: being i) acyclic, ii) key- and iii) foreign key-based [8, 14, 15, 20]. Examples will showcase how i) cyclic definitions result in ill-defined concepts and inefficient updates, ii) non-key functional dependencies cause data redundancy, inconsistency and inefficient operations, and iii) inclusion dependencies that are not foreign keys cause inefficient operations as well. We will further illustrate how Entity/Relationship diagrams can be obtained that are well-designed in this sense [6, 13].

There will also be a brief overview of recent approaches to apply conceptual modeling to graph databases [10–12, 17, 21], with restrictions on the order and depth of object types that are utilized. In contrast, the remaining tutorial will utilize the full expressiveness of Entity/Relationship models to play out their key strength for the benefit of graph databases [19].

Property Graphs and PG-Key. Next we introduce the concept of property graphs, illustrated on our running example as seen on the left of Fig. 2. We will then discuss the recent proposal for the definition of key constraints in property graphs, called PG-Key [3, 4]. This constitutes a powerful framework, and was designed by industry and academia. We will discuss the syntax and semantics of PG-Key expressions such that the audience will be comfortable understanding how expressive PG-Keys are and how they work. The discussion will be led by examples within the context of our running example.

PG-Schema. Next we will summarize PG-Schema [3], a recent proposal by academics and practitioners to help standardizing schema support for graph databases. The tutorial will survey the main goals, features, and provide several examples to illustrate PG-Schema definitions. These will be further used in subsequent sections of the tutorial, including PG-Key definitions. An important feature of PG-Schema was to "deliberately target minimal data modeling capabilities and as a reference point ... take the most basic variant of Entity-Relationship (ER) diagrams ... as the ultimate lower bound in the expressiveness of conceptual modelling languages" [3]. We will show that expressive E/R diagrams can be specified as PG-Schema, and constitute a great lower bound for PG-Schema as they capture well-designed property graphs.

Designing Graph Databases Well. The next two sections will constitute the main topics of interest for the tutorial. This section,

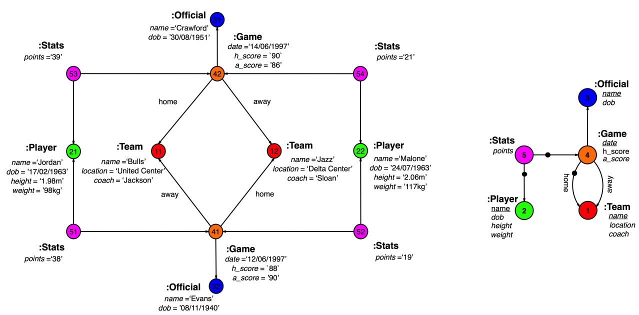
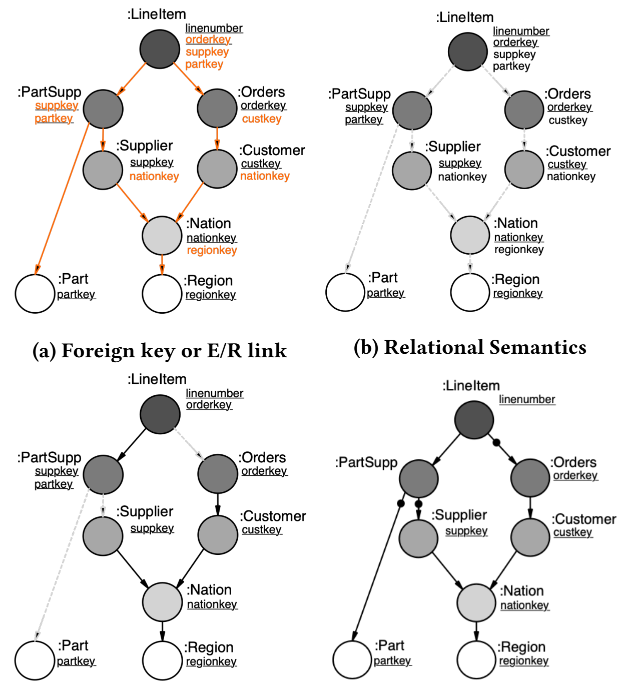


Figure 2: Mapping nodes of E/R graphs to their E/R diagram



(a) Foreign key or E/R link **(b) Relational Semantics**

(c) Mixed Semantics **(d) Graph Semantics**

Figure 3: E/R diagrams for TPC-H under different semantics

in particular, will show how traditional E/R modeling provides a methodology for designing property graphs well.

E/R Diagrams as Property Graphs, PG-Schema. Firstly, we will illustrate that E/R diagrams are property graphs themselves, but also constitute a fragment of PG-Schema. This situation is reminiscent of XML where XML Schema definitions constitute XML documents themselves [5]. We will outline the core ideas in formalizing E/R diagrams as property graphs and PG-Schema definitions, and illustrate the formalization on our running example. Indeed, the right of Fig. 2 shows the E/R diagram from Fig. 1 as a property graph. As a result, E/R modeling is available as a mature and trusted methodology for the design of property graphs.

E/R Graphs as Semantics of E/R diagrams. Secondly, we will showcase that E/R graphs constitute the first graph semantics for E/R diagrams, as defined in [19]. Compliance is formalized in the form of homomorphisms from the E/R graph on the E/R diagram. Instead of providing a formal definition, we will illustrate the homomorphisms on our running example using intuitive color codings. This is depicted in Fig. 2 where the homomorphism is defined by mapping nodes of the E/R graph on the left to nodes of the same color on the E/R diagram on the right. As E/R modeling is a conceptual approach and property graphs are considered as a logical data model, E/R graphs unify conceptual, logical, and graph data modeling. Hence, modern graph database systems provide an operational platform for conceptual data models, offering a viable alternative to relational technology without having to translate conceptual models at all.

Entity and Referential Integrity Management. Next we will demonstrate that E/R graphs and diagrams offer principled concepts to maintain entity and referential integrity efficiently within graph database systems. In particular, translations of relational databases to E/R graphs offer various benefits for data integrity management.

E/R Keys as Efficient Fragment of PG-Key. We will illustrate the notion of an E/R key and demonstrate that every E/R key is also a PG-Key, but not vice versa. We will further demonstrate that PG-Keys can express arbitrary keys and foreign keys from relational databases, when translated into property graphs. Because of that, the problem of deciding whether a given PG-Key is implied by a given set of PG-Keys is infeasible. However, implication of E/R keys can be decided in linear time. Hence, E/R keys form an efficient fragment of PG-Key, namely for well-designed property graphs. While graph database systems provide support for specifying and enforcing unique constraints, none of them can express E/R keys, but only the proper subclass of E/R keys that comprise properties but no components. We call this subclass *local keys.*

*Principled Semantics of E/R Graphs.*A goal was to represent well-designed databases as E/R graphs. In terms of the relational model of data, the class of well-designed databases is captured by schemata in Inclusion Dependency Normal Form (IDNF) [14, 15]. Hence, we will illustrate how to translate relational database schemata in IDNF into E/R diagrams, including their instances. We will show three principled ways in which these translations work. Firstly, we use a relational semantics in which we duplicate properties of a local key on nodes that reference them. Hence, this translation manages entity integrity by local keys and referential integrity by foreign keys, very much like relational databases. Secondly, we use a graph semantics in which we use E/R links instead of foreign keys, thereby replacing the need to duplicate foreign key properties by simply drawing a directed edge. Hence, entity integrity is managed by E/R keys while referential integrity is managed by E/R links. This is the first approach ever in which property redundancy is completely eliminated. As a compromise and driven by the fact that current graph database technology only supports local but not general E/R keys, we also showcase a mixed semantics. Here, properties are only duplicated on referencing nodes when they are required for a key. This ensures that entity integrity can be managed by local keys, that is, with current database technology. At the same time, we maximize the use of E/R links to manage referential integrity. This means we use either duplicated properties or directed edges whenever no duplication of properties is necessary. The different semantics will be showcased on our running example, but also on the TPC-H schema in Fig. 3 that we will present as a realistic use case. Fig. 3a shows for each non-leave node the choice between keeping the E/R link to another node or duplicating the key properties of the referenced node. Fig. 3b illustrates the use of relational semantics where we do not keep any E/R link, but only duplicate properties of the local key on the referenced node. In contrast, Fig. 3d shows the other extreme case where we always keep the E/R link and do not duplicate any property, while Fig. 3c illustrates the mixed semantics as a compromise.

Use Case of TPC-H as Well-designed Graph Database. Apart from the running example, we will present TPC-H as an industry-like use case that illustrates our main concepts and quantifies how well our E/R approach works at the operational level. Our choice is based on the TPC-H schema representing relationship types of higher order, a mix of local and E/R keys, and broad range of instances and operations. We will present translations of the schema into an E/R diagram using the different semantics, translate its operations, and compare the efficiency of their evaluation within Neo4j and MySQL. The results will demonstrate to the audience how well our initial research question has been addressed.

Summary and Open Problems. We will summarize our main findings and outline problems for future research. The findings have been mentioned under what the audience will learn. Future work will identify other fragments of PG-Schema and PG-Key with use cases for graph databases, build E/R diagram and E/R graph support in graph database systems, identify other classes of well-designed property graphs, transform property graphs into well-designed ones, or infer E/R diagrams from given property graphs.

Summary. The tutorial has a clear scope of providing some answer to the initial research question at a sufficient depth.

4 TUTORIAL

This chapter provides details about the tutorial.

Length. The tutorial will be delivered in 90 minutes. About 15 minutes will be spent on E/R modeling, and 15 minutes on PG-Key and PG-Schema. Motivation, goals and summary will take about 15 minutes together. We will spend 45 minutes on the design of graph databases and integrity management.

Style. The tutorial will provide overviews of conceptual and graph modeling languages but also show how to combine them. The style will be inquisitive and illustrative, leaving time for questions and discussions. Lots of examples and visualizations will illustrate ideas, techniques and findings.

Presenters. Philipp is a research fellow and teaching assistant with experience teaching database topics to undergraduate and postgraduate students of different backgrounds. The topic of his PhD was on the design of graph databases, with publications in VLDB, SIGMOD, and VLDBJ. Sebastian is a Professor of Computer Science with vast experience in presenting database topics to audiences of various backgrounds. He has presented extensively at top database conferences, particularly on topics of the tutorial.

Prior Offering. The tutorial has not been offered before.

REFERENCES

[1] Renzo Angles, Marcelo Arenas, Pablo Barceló, Peter A. Boncz, George H. L. Fletcher, Claudio Gutierrez, Tobias Lindaaker, Marcus Paradies, Stefan Plantikow, Juan F. Sequeda, Oskar van Rest, and Hannes Voigt. 2018. G-CORE: A Core for Future Graph Query Languages. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. 1421–1432.

[2] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan L. Reutter, and Domagoj Vrgoc. 2017. Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv.* 50, 5 (2017), 68:1–68:40.

[3] Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savkovic, Michael Schmidt, Juan Sequeda, Slawek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoc, Mingxi Wu, and Dusan Zivkovic. 2023. PG-Schema: Schemas for Property Graphs. *Proc. ACM Manag. Data* 1, 2 (2023), 198:1–198:25. https://doi.org/10.1145/3589778

[4] Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Keith W. Hare, Jan Hidders, Victor E. Lee, Bei Li, Leonid Libkin, Wim Martens, Filip Murlak, Josh Perryman, Ognjen Savkovic, Michael Schmidt, Juan F. Sequeda, Slawek Staworko, and Dominik Tomaszuk. 2021. PG-Keys: Keys for Property Graphs. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. 2423–2436. https://doi.org/10.1145/3448016.3457561

[5] Geert Jan Bex, Frank Neven, and Jan Van den Bussche. 2004. DTDs versus XML Schema: A Practical Study. In *Proceedings of the Seventh International Workshop on the Web and Databases, WebDB 2004, June 17-18, 2004, Maison de la Chimie, Paris, France, Colocated with ACM SIGMOD/PODS 2004*. 79–84.

[6] Joachim Biskup, Ralf Menzel, Torsten Polle, and Yehoshua Sagiv. 1996. Decomposition of Relationships through Pivoting. In *Conceptual Modeling - ER'96, 15th International Conference on Conceptual Modeling, Cottbus, Germany, October 7-10, 1996, Proceedings*. 28–41.

[7] Angela Bonifati, George H. L. Fletcher, Hannes Voigt, and Nikolay Yakovets. 2018. *Querying Graphs*. Morgan & Claypool Publishers.

[8] Peter P. Chen. 1976. The Entity-Relationship Model - Toward a Unified View of Data. *ACM Trans. Database Syst.* 1, 1 (1976), 9–36.

[9] Peter P. Chen. 1997. English, Chinese and ER Diagrams. *Data Knowl. Eng.* 23, 1 (1997), 5–16.

[10] Gwendal Daniel, Gerson Sunyé, and Jordi Cabot. 2016. UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases. In *Conceptual Modeling - 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings*. 430–444.

[11] Victor Martins de Sousa and Luís Mariano del Val Cura. 2018. Logical Design of Graph Databases from an Entity-Relationship Conceptual Model. In *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services, iiWAS 2018, Yogyakarta, Indonesia, November 19-21, 2018*. 183–189.

[12] Ewout Gelling, George Fletcher, and Michael Schmidt. 2023. Bridging graph data models: RDF, RDF-star, and property graphs as directed acyclic graphs. *CoRR* abs/2304.13097 (2023).

[13] Sven Hartmann. 2001. Decomposing relationship types by pivoting and schema equivalence. *Data Knowl. Eng.* 39, 1 (2001), 75–99.

[14] Mark Levene and Millist W. Vincent. 2000. Justification for Inclusion Dependency Normal Form. *IEEE Trans. Knowl. Data Eng.* 12, 2 (2000), 281–291.

[15] Heikki Mannila and Kari-Jouko Räihä. 1992. *Design of Relational Databases*. Addison-Wesley.

[16] Wim Martens, Frank Neven, Matthias Niewerth, and Thomas Schwentick. 2017. BonXai: Combining the Simplicity of DTD with the Expressiveness of XML Schema. *ACM Trans. Database Syst.* 42, 3 (2017), 15:1–15:42. https://doi.org/10.1145/3105960

[17] Jaroslav Pokorný. 2016. Conceptual and Database Modelling of Graph Databases. In *Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS 2016, Montreal, QC, Canada, July 11-13, 2016*. 370–377.

[18] Sherif Sakr, Angela Bonifati, Hannes Voigt, Alexandru Iosup, Khaled Ammar, Renzo Angles, Walid G. Aref, Marcelo Arenas, Maciej Besta, Peter A. Boncz, Khuzaima Daudjee, Emanuele Della Valle, Stefania Dumbrava, Olaf Hartig, Bernhard Haslhofer, Tim Hegeman, Jan Hidders, Katja Hose, Adriana Iamnitchi, Vasiliki Kalavri, Hugo Kapp, Wim Martens, M. Tamer Özsu, Eric Peukert, Stefan Plantikow, Mohamed Ragab, Matei Ripeanu, Semih Salihoglu, Christian Schulz, Petra Selmer, Juan F. Sequeda, Joshua Shinavier, Gábor Szárnyas, Riccardo Tommasini, Antonino Tumeo, Alexandru Uta, Ana Lucia Varbanescu, Hsiang-Yun Wu, Nikolay Yakovets, Da Yan, and Eiko Yoneki. 2021. The future is big graphs: a community view on graph processing systems. *Commun. ACM* 64, 9 (2021), 62–71. https://doi.org/10.1145/3434642

[19] Philipp Skavantzos and Sebastian Link. 2025. Entity/Relationship Graphs: Principled Design, Modeling and Data Integrity Management of Graph Databases. *Proc. ACM Manag. Data* 3, 3 (2025), 1–25.

[20] Bernhard Thalheim. 2000. *Entity-relationship modeling - foundations of database technology*. Springer.

[21] Roberto De Virgilio, Antonio Maccioni, and Riccardo Torlone. 2014. Model-Driven Design of Graph Databases. In *Conceptual Modeling - 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014. Proceedings*. 172–185.