

Building Executable Entity-Relationship Models

A Hands-On Tutorial for Understanding Sustainability in Supply Chains Using TypeDB

Joshua Send¹

¹TypeDB, Sas House Friarswood, Chipperfield Road, Kings Langley, Hertfordshire, WD4 9JB, United Kingdom

Abstract

Fifty years on, Chen’s Entity-Relationship model is still translated into poorer implementation models—losing relationship identity, subtypes, roles, and participation constraints along the way. This tutorial executes a conceptual model directly in TypeDB, whose language TypeQL implements a higher-level polymorphic entity-relation-attribute model grounded in type theory (Best Newcomer Award, ACM SIGMOD/PODS 2024). Building a sustainable supply-chain model, participants progress from a classic ER baseline to polymorphic querying, n-ary supplier networks, and recursive traceability—and leave able to turn an ER diagram into a running, schema-validated database.

Keywords

Entity-Relationship model, conceptual modeling, executable schemas, TypeDB, TypeQL, polymorphic query languages, supply chain traceability

1. Presenter

Joshua Send — CTO, TypeDB. Contact: joshua@typedb.com.

Joshua is the CTO of TypeDB. He has been at the company for seven years, helping lead the rewrite of TypeDB 2 into Rust as well as designing and implementing the core TypeQL feature set. He obtained his bachelor’s and master’s degrees in computer science at the University of Cambridge. He has previously presented to the Rust NYC and LinkML communities.

2. Goals and Objectives

Overall Goal. Enable participants to take a conceptual model from diagram to running, schema-validated database with substantially less semantic loss than classical implementation paths—and to understand precisely which semantics standard ER-to-relational and graph mappings lose, so that they can evaluate executable conceptual modeling on its merits. To this end, the tutorial introduces TypeDB as both a specification *and* querying tool for data models.

Concrete Objectives. By the end of the tutorial, participants will be able to:

1. Identify the specific semantic losses of classical mappings — relationship identity, subtype/generalization constraints, roles, and participation constraints [1];
2. Map Chen’s ER and EER constructs onto TypeQL construct-by-construct, including where the correspondence is exact and where it departs;
3. Define a TypeQL schema for a supply chain domain, expressing subtype hierarchies, n-ary multi-role relations, and cardinality/participation constraints directly;
4. Write and run polymorphic queries against that schema, including recursive traceability queries over multi-tier supplier networks with real deforestation-risk data;

ER2026: Companion Proceedings of the 45th International Conference on Conceptual Modeling: Doctoral Consortium, Tutorials, Posters and Demos, October 5-8, 2026, St. John’s, NL, Canada.

✉ joshua@typedb.com (J. Send)



Copyright © 2026 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

5. Choose the appropriate level of schema strictness for a given application—from label-based flexibility to fully enumerated, validated types—and articulate the trade-offs;
6. Position executable conceptual modeling relative to relational, property graph, and RDF/OWL approaches.

3. Audience

Intended Audience. General: researchers, graduate students, educators, and practitioners interested in conceptual modeling and in implementing ER models as running systems.

Level. Basic/introductory. No prior knowledge of TypeDB or TypeQL is assumed.

Prerequisites. Familiarity with basic ER modeling concepts, and general technical familiarity (e.g., connecting to a server). Participants who wish to follow the hands-on portion on their own machines should bring a laptop with a modern web browser.

4. Relevance and Novelty

This tutorial introduces a modeling tool that allows anyone to accelerate building well-designed, elegant knowledge bases and schemas, while remaining grounded in the conceptual-modeling tradition of the ER community.

Fit With ER's Scope. TypeDB implements a polymorphic entity-relation-attribute (PERA) model [2], close in intent and usage to the original ER model proposed by Chen [3]. The tutorial directly addresses a long-standing concern of the community: the semantic information lost when conceptual models are translated into poorer implementation models [1]. It can further be seen as a production-grade step toward the community's own vision of conceptual schemas that are themselves executable [4, 5], in the lineage of executable conceptual languages such as Taxis [6] and Galileo [7].

Relevance to Practice. TypeDB is a growing, modern, open-source database for managing complex datasets, and is ready for production use today. It has been used in published academic work to build knowledge representation systems in domains including archival research, robotics, and space systems. The techniques taught transfer directly to participants' own domains.

Distinctiveness for ER Participants. The tutorial demonstrates new ideas for addressing questions ER modeling was originally designed to answer—in particular, avoiding the loss of semantic information between model and implementation—by reducing the gap between the conceptual model and its executable form to near zero. The hands-on component uses real open supply-chain sustainability data, aligning with the ER 2026 theme.

5. Projected Benefits

Participants gain hands-on experience with a database designed to alleviate the burden of building and iterating on executable conceptual models. Concretely, they leave with: (i) the ability to turn an ER diagram into a running, schema-validated, queryable database; (ii) a construct-by-construct mapping from ER/EER to TypeQL they can apply to their own models; (iii) a clear picture of what classical translations lose and how to weigh executable conceptual modeling against relational, property graph, and RDF/OWL alternatives; and (iv) all tutorial materials (schema, dataset, and exercises), published openly and available after the conference.

6. Detailed Outline and Timetable

The tutorial is designed for a 90-minute slot.

Table 1

Tutorial segments and timing

Segment	Content	Min.
<i>Part 1 — Lecture (30 min)</i>		
1.1	Introduction; TypeDB and its source of inspiration: Chen’s ER model	10
1.2	Translations of ER (relational, graph) and their semantic losses	10
1.3	The PERA model; mapping ER constructs to TypeQL, incl. departures	10
<i>Part 2 — Guided hands-on tutorial (55 min)</i>		
2.1	Setup: browser-based environment; whiteboarding the initial ER model	10
2.2	Translating the model into a TypeDB schema, with explanation	10
2.3	Loading data; first queries	10
2.4	Polymorphic and recursive queries: supplier networks, traceability	15
2.5	Extending the model live; inspecting the data model by querying it	10
3	Wrap-up: comparative positioning, open problems, discussion	5

7. Tutorial Method

Teaching Approach. Guided hands-on modeling, with challenges interspersed. Part 1 is a lecture with worked examples. In Part 2 the presenter builds the supply-chain model live in define–insert–query cycles, which participants can follow in a browser-based TypeDB environment; segments end with a short challenge query that participants attempt before the walkthrough (e.g., “find all products with tier-3 deforestation exposure”). Questions are taken throughout.

Technology Requirements. Beyond standard AV, only a projector is required for the presentation. Participants following along hands-on need a laptop with a web browser and conference Wi-Fi; a browser-based environment is provided, with no installation required. All materials remain available online after the tutorial.

8. Prior Venues

This tutorial has not been presented before; this is the first presentation of this material.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Code to: generate LaTeX, iterate on prose, and grammar and spell check. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

- [1] T. J. Teorey, D. Yang, J. P. Fry, A logical design methodology for relational databases using the extended entity-relationship model, *ACM Computing Surveys* 18 (1986) 197–222.
- [2] C. Dorn, H. Pribadi, Typeql: A type-theoretic & polymorphic query language, *Proceedings of the ACM on Management of Data* 2 (2024) Article 110.

- [3] P. P.-S. Chen, The entity-relationship model—toward a unified view of data, *ACM Transactions on Database Systems* 1 (1976) 9–36.
- [4] A. Olivé, Conceptual schema-centric development: A grand challenge for information systems research, in: *CAiSE 2005*, volume 3520 of *LNCS*, Springer, 2005, pp. 1–15.
- [5] D. W. Embley, S. W. Liddle, O. Pastor, Conceptual-model programming: A manifesto, in: D. W. Embley, B. Thalheim (Eds.), *Handbook of Conceptual Modeling*, Springer, 2011, pp. 3–16.
- [6] J. Mylopoulos, P. A. Bernstein, H. K. T. Wong, A language facility for designing database-intensive applications, *ACM Transactions on Database Systems* 5 (1980) 185–207.
- [7] A. Albano, L. Cardelli, R. Orsini, Galileo: A strongly-typed, interactive conceptual language, *ACM Transactions on Database Systems* 10 (1985) 230–260.