



Chapter I

Enforcing Cardinality Constraints in the ER Model With Integrity Methods¹

Mira Balaban and Peretz Shoval
Ben-Gurion University of the Negev, Israel

Entity-Relationship (ER) schemas include cardinality constraints that restrict the dependencies among entities within a relationship type. The cardinality constraints have direct impact on application transactions, since insertions or deletions of entities or relationships might affect related entities. Application transactions can be strengthened to preserve the consistency of a database with respect to the cardinality constraints in a schema. Yet, once an ER schema is translated into a logical database schema, the direct correlation between the cardinality constraints and application transaction is lost, since the components of the ER schema might be decomposed among those of the logical database schema.

We suggest extending the Enhanced-ER (EER) data model with integrity methods that can enforce the cardinality constraints. The integrity methods can be fully defined by the cardinality constraints, using a small number of primitive update methods, and are automatically created for a given EER diagram. A translation of an EER schema into a logical database schema can create integrity routines by translating the primitive update methods alone. These integrity routines may be implemented as database procedures, if a relational DBMS is utilized, or as class methods, if an object-oriented DBMS is utilized.

INTRODUCTION

The Entity-Relationship (ER) data model was introduced by Chen (1976) as a means for describing in a diagrammatic form, entities and relationships among entities in the subject domain. The ER model enjoys widespread popularity as a tool for conceptual database design, and received many extensions and variations, which are generally termed the Enhanced-ER (EER) model. An EER schema can be translated into logical database schemas, usually relational, and implemented with some specific DBMS, using its specific data

definition language (DDL). Application programs that manipulate the database access the DBMS via its data manipulation language (DML), either directly or through a host programming language.

EER can be used not only to design a conceptual schema that will later on be translated into a logical schema, but also as a platform for database integration, i.e., to create a meta-schema for a multi database environment in which there are heterogeneous databases, utilizing different data models. Cooperation or federation of such databases is possible if a common meta-schema is created. EER can be the high-level model used for that purpose. Similarly, the EER model is used in database re-engineering; the data model of a legacy-system is first reverse-engineered to an EER schema, and later on translated and implemented in a new DBMS. Yet, the EER model deals only with the static (structural) aspects of the data model (namely, entities, relationships and attributes), but not with behavioural aspects (namely, procedures to manipulate the data that is defined by the schema, and to preserve the integrity of data). These aspects are taken care of at the implementation level, either by the DBMS (for example, when a relational DBMS performs referential integrity checks), or by the application programs.

An EER schema supports the specification of cardinality constraints, which restrict the dependencies among entities within a relationship type (see, for example, Lenzerini & Santucci, 1983; Lenzerini & Nobili, 1990; Ferg, 1991; Thalheim, 1992; Thalheim, 1998; Balaban & Shoval, 2001). For example, the cardinality constraints can specify that a department must have at least five workers and at most two hundred; or that an employee must work for one department only. The cardinality constraints have direct impact on maintenance transactions of the target system, since insertions or deletions of entities or relationships might affect related entities. This impact can be captured by operations that a transaction must trigger in order to preserve the cardinality constraints. Yet, once an EER schema is translated into a logical database schema, the direct correlation between the cardinality constraints and maintenance transactions is lost, since the components of the EER schema are usually decomposed among those of the target database schema. Moreover, at this level it is up to application programmers to correctly capture the constraints.

In this chapter, we suggest to enhance the EER model with the behavioural aspects of the cardinality constraints (see also Lazarevic & Masic, 1991; Balaban & Shoval, 2001). Specifically, we suggest extending the EER data model with *integrity methods*, i.e., methods that maintain the consistency of data according to the schema definition. Maintenance of consistency means that any attempt to add, delete or change entities and relationships in the subject domain of a schema is checked against the schema definitions, as expressed by cardinality constraints. The integrity methods can be fully defined by the cardinality constraint. Hence, once the enhanced EER schema is mapped to a logical schema, the integrity methods can be mapped to respective integrity routines. These integrity routines may be implemented as database procedures, if a relational DBMS is utilized, or as class methods, if an object-oriented DBMS is utilized. The integrity methods are built on top of *primitive update methods* that perform the update transactions. This separation adds a layer of abstraction that enables the mapping of a behavior-enhanced EER schema into some target database schema, in terms of the primitive methods alone.

Another popular data model, which is closely related to the EER model, is the Object-Oriented (OO) model. The OO model supports static data abstraction using *object classes*, and models system behavior through *methods* (procedures) that are attached to object classes, with message passing as a means for communication among objects. Moreover, unlike EER, which is mainly used for conceptual data modeling, the OO approach is also

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/enforcing-cardinality-constraints-model-integrity/4319

Related Content

Analysis of Key Barriers in Blockchain in Banking: ISM Ranking Approach

Gargi Pant Shukla and Nitin Balwani (2022). *Applications, Challenges, and Opportunities of Blockchain Technology in Banking and Insurance* (pp. 83-98).

www.irma-international.org/chapter/analysis-of-key-barriers-in-blockchain-in-banking/306456

On the Adaptation of an Agile Information Systems Development Method

Mehmet N. Aydin, Frank Harmsen, Kees van Slooten and Robert A. Stagwee (2005). *Journal of Database Management* (pp. 25-40).

www.irma-international.org/article/adaptation-agile-information-systems-development/3340

Using Temporal Versioning and Integrity Constraints for Updating Geographic Databases and Maintaining Their Consistency

Wassim Jaziri, Najla Sassi and Dhouha Damak (2015). *Journal of Database Management* (pp. 30-59).

www.irma-international.org/article/using-temporal-versioning-and-integrity-constraints-for-updating-geographic-databases-and-maintaining-their-consistency/140545

A Big Data Application for Energy Consumption Management of Commercial Buildings

Sarah Abdullah Al Muhaysh, Thanaa Saad Al Salem, Fatimah Abdullah Al Jaafari, Al Anoud Saleem Al Shaikh Hussain and Amir Ali Khwaja (2021). *International Journal of Big Data Intelligence and Applications* (pp. 1-20).

www.irma-international.org/article/a-big-data-application-for-energy-consumption-management-of-commercial-buildings/287615

Abstract DTD Graph from an XML Document: A Reverse Engineering Approach

Joseph Fong and Herbert Shiu (2010). *Principle Advancements in Database Management Technologies: New Applications and Frameworks* (pp. 204-224).

www.irma-international.org/chapter/abstract-dtd-graph-xml-document/39357