In Part I, this chapter surveys the state of the art of the semantic integrity constraints in some relational and object relational available database systems. In Part II, it also provides an overview of the SQL standard integrity issues and describes semantic integrity support in the following DBMSs: Oracle, IBM DB2, Informix, Sybase and PostgreSQL.

The major differences and similarities among these systems are analyzed in relation to the definition, semantics and fidelity to the SQL standard prescriptions.

PART I: INTRODUCTION

This chapter is devoted to expanding the concepts presented in Chapter I. One of the most important current trends in database management is the increase of the semantic content of stored data. In this way, the first step in the establishment of the database theory is the precise definition of data models, since without it the database concepts cannot be understood as regards the
design, analysis and implementation of schemas, transactions, and databases (Thalheim, 1996).

Taking into account that a database is a resource shared by many applications, it is advisable to register any knowledge about data semantics in the database in such a way that there is no need to replicate it into the applications using such knowledge. This knowledge covers a large variety of fields of the Universe of Discourse (UofD) “under the form” of rules, which can be grouped in the following families: rules about the valid values of particular items of data; rules describing the way the data are associated with one another (interdata connections); and rules about the actions that should be performed when a specific event shows up (business or enterprise rules). Generally, the first two kinds of rules are included under the denomination “integrity constraints.” However, the concept of rules is preferable since, in general, the distinction between constraints and business rules is not clear. From an operational point of view, all rules can be treated as active requirements since the system must verify that user manipulations leave the data in an allowed state. If the execution of a proposed transaction leads to a constraint violation, the system either aborts the transaction or executes repairing actions, clearly revealing a reactive nature.

In the database world, rules are relevant concepts to describe a piece of active requirements. Rules define the intended structural and behavioral properties of objects involved in a database application, and they can be specified in several ways. At procedural and production levels, rules clearly exhibit a reactive structure. At the conceptual level, some rules already have an active form while some others do not, but all of them involve active requirements (Van den Berghe, 1999).

When the database engine automatically enforces rules like these, stored data become more “active,” thus acquiring a richer level of semantic content (Chamberlin, 1998). In other words, database constraints can be regarded as a language to specify the semantics of data.

Most database systems provide some support for integrity constraints. For example, current commercial database systems (especially RDBMSs) enforce only a small set of constraints, mainly because of the performance overhead associated with update operations. In this manner, in RDBMSs and ORDBMSs some restrictions related to the valid values of a particular column (typing constraints) can be directly represented at schema definition time using the facilities the language (usually SQL) offers for the data definition (DDL). Others are expressed and enforced by mechanisms such as check conditions, assertions and triggers in RDBMSs or specific methods in
An Integrated Query Relaxation Approach Adopting Data Abstraction and Fuzzy Relation


www.igi-global.com/article/integrated-query-relaxation-approach-adopting/47419?camid=4v1a