Chapter VII

Normalization of Relations with Nulls in Candidate Keys: Traditional and Domain Key Normal Forms

George C. Philip, University of Wisconsin Oshkosh, USA

ABSTRACT

This chapter discusses normalization of relations when the candidate keys of a relation have missing information represented by nulls. The chapter shows that problems and confusion can arise in normalizing relations with nulls in candidate keys. Candidate keys with missing information commonly are found in relations that represent information on two entities with a one-to-one relationship between them. The current definition of Boyce-Codd Normal Form (BCNF) is ineffective in identifying poor designs in such relations that may have insertion/deletion anomalies. Domain Key Normal Form (DKNF) also suffers from the same problem. It is shown that the above problem can be corrected by incorporating the concept of entity integrity rule into the definitions of BCNF and DKNF. This chapter also shows that incorporating the entity integrity rule into the definition of either a relation or a candidate key does not provide a satisfactory solution to the problem.
INTRODUCTION

The relational database design concepts were developed without considering missing information in relations (Codd, 1986; Levene, 1999; Date 2000). Value of an attribute in a tuple may be missing for several reasons: 1) Value is applicable but it is unknown, 2) Value is not applicable, 3) Value does not exist, or 4) Other reasons, such as value is undefined (Date 2000). An example of value that is not applicable is the attribute driver license number for a ten-year-old child. If an adult does not have a driver license number, then the value does not exist. If an adult has a driver license number, but it is unknown, then the value is applicable but unknown.

A common method of representing missing values is using nulls (Codd, 1986). Other methods include using default values (Date, 1990), using a subset of the attribute domain (Lipski, 1979), and using variables or many different “null values” (Imielinski & Lipski, 1984). Missing information can create problems in querying data from relations (Imielinski & Lipski, 1984; Date, 1990). Several methods have been proposed to extend the relational operators to deal with missing values (Codd, 1986; Reiter, 1986; Sutton & King, 1995). Another group of studies examined the effect of nulls on the concept of functional dependency (Vassiliou, 1980; Vardi, 1986; Levene & Loizou, 1999). These studies have focussed primarily on missing values of the type “applicable but unknown”.

The current paper examines the effect of nulls in candidate keys on normalizing a relational schema. The nulls considered in this paper are of the type “not applicable” or “does not exist”. Specifically, this paper examines the effectiveness of Boyce-Codd Normal Form (BCNF) and Domain Key Normal Form (DKNF) in identifying insertion/deletion anomalies if missing values in candidate keys are represented by nulls. Candidate keys with nulls commonly are found in relations that represent information on two entities with a one-to-one relationship between them. It is shown that the current definition of Boyce-Codd Normal Form is ineffective in identifying poor designs in such relations. Domain Key Normal Form (DKNF) also suffers from the same problem. The paper identifies the source of the problem and offers a solution by incorporating the concept of entity integrity rule into the definitions of BCNF and DKNF. This paper also shows that incorporating the entity integrity rule into the definition of either a relation or a candidate key does not provide a satisfactory solution to the problem.

DESCRIPTION OF THE PROBLEM

To help explain the problem, we consider two entities, EMPLOYEE and COMPUTER, that have a (zero-or-one)-to-(zero-or-one) relationship between them. Thus, a computer has zero or one employee assigned to it at any given time. Similarly, an employee is assigned to zero or one computer at any time. Consider a relation:

ASSIGNMENT (ID, NAME, TITLE, COMPUTER_NO, MODEL, RAM).

In the above relation, ID, NAME, and TITLE represent the identification number, the name, and the title of the employee, respectively. ID is the only unique identifier of the employee. COMPUTER_NO is the only unique identifier of the computer assigned to the employee. MODEL and RAM represent the model, and the amount of memory of the employee’s computer, respectively. Figure 1 shows a sample state of the relation.
Related Content

Bind but Dynamic Technique: The Ultimate Protection Against SQL Injections
www.igi-global.com/chapter/bind-dynamic-technique/20774?camid=4v1a

A Web-Based Application to Exchange Electronic Health Records and Medical Images in Ophthalmology
www.igi-global.com/chapter/web-based-application-exchange-electronic/7978?camid=4v1a

Hypermedia Document Management: A Metadata and Meta-Information System
www.igi-global.com/article/hypermedia-document-management/3262?camid=4v1a
Deciding Query Entailment in Fuzzy OWL Lite Ontologies
www.igi-global.com/chapter/deciding-query-entailment-fuzzy-owl/52304?camid=4v1a