THEORETICAL ASPECTS OF MODELING IN RELATIONAL DATA BASE

(6)

C. DELOBEL

RR N° 177 JULY 1979

Résumé : Le but de ce rapport est de faire une synthèse des principaux résultats obtenus dans le développement de la théorie du modèle relationnel de base de données. Les principaux concepts sont passés successivement en revue : le concept de dépendance sous ses différentes formes, fonctionnelle, multivaluée, hiérarchique et produit, le concept de formes normales d'une relation, et enfin le processus de conception des schémas relationnels de base de données. Les principaux problèmes à résoudre comme la dérivation des dépendances, la recherche de la clé d'une relation, l'équivalence entre schémas, et la construction de schéma dans une forme donnée sont présentés ainsi que les principales techniques et algorithmes qui permettent de les solutionner.

Abstract : The gôal of this paper is to present an overview of the theoretical background used in the relational data model. The basic notions such as : the concept of data dependencies, the concept of normal forms, and the process of designing relational schema are presented. Formal manipulation on data dependencies, the membership problem, the key finding problem and equivalence between different relational schema are introduced with the basic techniques used to solve them. Different algorithms for the design of relational schema are given.

1. INTRODUCTION[1]

The purpose of this paper is to present an overview of the theoretical background used in the relational data model. This presentation is divided essentially into four parts :

(a) Data dependencies

In the relational data model semantic properties about data has concentrated on dependency constraints. We shall present an overview of the different kind of dependencies : functional, multivalued, hierarchical and mutual. The properties of these dependencies will be studied and the concept of axiomatization will be introduced. Formal manipulation on data dependencies will be related to manipulation on boolean functions.

(b) Relational schema design

Data dependencies play an important role in the area of schema design since the definitions of data dependencies are related to the problem of how to decompose a given relation into two sub-relations (or more) so that the original one can be regenerated by the composition operation of the two sub-relations. The problem of schema design can be summarized as the problem : given an initial schema, find an equivalent one that is better in some respect. Several criteria for one schema to be "better" that another will be presented. As we will see different definitions of "better" and "equivalent" lead basically to two schema design methods : the synthesis approach and the decomposition approach.

(c) The synthesis approach

The synthesis approach is based essentially on functional dependencies. We will present an overview of the different algorithms for finding the closure and a covering of functional dependencies, and for building a relational schema according to a different degree in Codd normal form.

(d) The decomposition approach

The decomposition approach is based on the fact to apply recursively the decomposition of a relation into sub-relations. We will present the drawbacks of this method. It will be interesting to relate these problem to the definitions of views in a relational data base system.

---

[1] *Some of the materials used to prepare this synthesis have been extracted from papers written by myself or in cooperation with S. Parker, chapter 4 is extracted from [PD 79].*

A constraint can be seen as an intrinsic property of the data, for example, suppose that parts in an inventory are described by a relation R(PART-NUMBER, COLOR, PRICE, ...). A priori, any relation of this form can exist in the database. However if one specifies a constraint : the PRICE must range between $0 and $100 a piece, then only relations in which this constraint is valid can exist in the database. Similarly the specification that the knowledge of the PART-NUMBER-value implies the COLOR-value is also another type of constraint. In this case, this type of constraint is called a functional dependency and denoted PART-NUMBER → COLOR.

In this paper, we are investigating only some constraints, the family of dependencies : functional dependency, multivalued dependency, hierarchical dependency, join dependency, mutual dependency, which are presented in section 3.

Let Σ be a set of dependencies, we shall say that a relation R defined over the attributes $U = \{a_1, a_2, \ldots, a_n\}$ is of *type* or an *instance* of the *relational schema* $\underline{R} = \langle U, \Sigma \rangle$ iff each dependency $\sigma \in \Sigma$ holds in R.

Let $\underline{R} = \langle U, \Sigma \rangle$ be a relational schema, when we say that Σ logically implies σ (in the context of U), or that σ is a logical consequence of Σ (in the context of U), we mean that whenever Σ holds for a relation with attributes U, then so does σ. That is, there is no "counter example relation" R with attributes U such that every dependency in Σ hold in R, but such that σ does not hold in R. We write $\Sigma \models_U \sigma$, or if the context is not necessary $\Sigma \models \sigma$.

It is convenient to consider than the set of dependencies is closed under logical consequence. So that we can always speak of the dependencies in the schema, rather than dependencies that are logical consequence of those in the schema. That is, if Σ is a set of dependencies in the schema, and if σ is a dependency such that $\Sigma \models \sigma$, then σ is also a dependency in the schema.

2.2. Operations on relations

In his original presentation of the relation model [Cod 70], Codd introduced the relational algebra as a data manipulation language. There are two basic operations that will be of some interest of us : projection and natural join.