Inclusion Dependencies

Laura C. Rivero
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina and
Universidad Nacional de La Plata, Buenos Aires, Argentina

INTRODUCTION

Inclusion dependencies support an essential semantics of the standard relational data model. An inclusion dependency is defined as the existence of attributes (the left term) in a table R whose values must be a subset of the values of the corresponding attributes (the right term) in another table S (Abiteboul, Hull & Vianu, 1995; Codd, 1990; Connolly, Begg & Strachan, 1999; Date, 2000). When the right term conforms a unique column or a primary key (K) for the table S, the inclusion dependency is key-based (also named referential integrity restriction, *rir*). In this case, the left term is a foreign key (FK) in R and the restriction is stated as R[FK] <<S[K]. On the contrary, if the right term does not constitute the key of S, the inclusion dependency is non-key-based (simply, an inclusion dependency, *id*). Ids are expressed as $R[X] \subseteq S[Z]$, being R[X] and S[Z] the left and right terms respectively. Both, *rir*s and *id*s, are often called referential constraints.

*Rir*s and referential actions are important because they contain basic local semantic aspects, which have been elicited from the Universe of Discourse (UofD). They are sufficient to symbolize many natural semantic links such as the relationships and hierarchies that are captured by semantic models (Abiteboul, Hull & Vianu, 1995). In a different scenario, late changes of the logical design, disregarding the conceptual design, usually promotes some denormalization degree or the presence of complex n-ary relationship constructs. The decomposition and the synthesis coming from that process may add other restrictions, frequently adopting the form of *id*s that misrepresent objects and the corresponding inter-object relationships.

*Rir*s can be declaratively defined via the SQL foreign key clause (SQL:1999-1, 1999, SQL:1999-2, 1999) and are enforced by most current database systems:

FOREIGN KEY (<referencing column list>) REFERENCES <referenced table name> [(<referenced column list>)]
[MATCH <match type>]
[ON UPDATE <update referential action>]
[ON DELETE <delete referential action>]

The *rir*s can be specified with respect to different match types: SIMPLE (implicit if no match option is declared), PARTIAL and FULL. As it has been stated in the SQL:1999 standard document (SQL:1999-2, 1999): If <match type>(SIMPLE) is not specified, then for each row in the referencing table, either the referencing column has at least one null value or its value matches the value of a corresponding row in the referenced table. If PARTIAL is specified, then for each row in the referencing table the value of each foreign key column is null, or it has at least one non-null value that equals the corresponding referenced column value. Finally, if FULL is specified, for each row in the referencing table, either all foreign key columns have been instanced with null or the foreign key value equals the value of the corresponding referenced column.

When an integrity restriction is violated, the usual response of the system is the rollback of the data manipulation intended by the user. In the case of *rir*s, some other alternative actions are possible. These actions, named referential actions or referential rules, specify the behavior of the left and right relations under the deletion or the updating of a referenced row (in the right table), or the insertion in the referencing (left) table. Possible actions are: cascade, restrict, no action, set null, set default (Date, 2000; Markowitz, 1994; SQL:1999_2, 1999; Türker & Gertz, 2001).

With the cascade option, the referencing rows will be deleted (updated) together with the referenced row. With the set null (set default) option, all references to the deleted (updated) row will be set to null (default) values. Restrict and no action rules disallow the deletion (update) of the referenced row, if there exists rows in the left table referencing it. The unique referential rule for insertions is restrict: inserting a row into the referencing table is possible only if the referenced tuple already exists in the right term.

On the other hand, *id*s are usually defined with check statements

CHECK (<referencing column list> IN (SELECT <referenced column list> FROM <referenced table>))

or triggers, thus complicating the development of application programs and integrity maintenance (Connolly, Begg & Strachan, 1999; Date, 2000; Date & Darwen, 1997; Elmasri & Navathe, 2000).

The objective of this entry is to provide an overview of *id*s, summarizing main research on this topic.

BACKGROUND

The comprehension of the syntactic and semantic issues related to referential restrictions is facilitated by the study of the structure of their terms.

Structure

Considering a relation shape, there are five possible placements of a non-empty set of attributes with regard to the key placement. Being W such set of attributes, and K the primary key of R, the five placements are depicted in Figure 1: I) W≡ K (W coincides with K); II) W≡ Z, being Z a subset of non-key attributes (W and K have not common attributes); III) W≡ K_1, being K_1 a proper subset of K, $K_1 \neq \emptyset$ (the set of attributes in W is a proper subset of the set of attributes in K); IV) W≡ K ∪ Z (W attributes include the complete set of attributes of K); and finally V) W≡ $K_1 \cup$ Z, $K_1 \neq \emptyset$ (W and K partially overlap). In all cases, Z ≠ ∅.

Let R and S be the left and right terms of a referential constraint, respectively. Taking into account the placements in Figure 1, 25 pairs <R[W_R], S[W_S]> corresponding to left and right terms of a referential constraint can be derived. The five cases having S[W_S] as the primary key for S (numbered 1 to 5 in Table 1) correspond to *rir*s.

Semantic Perspective

*Rir*s of types I, II and III represent typical relationships in semantic models (Abiteboul, Hull & Vianu, 1995). Type I depicts subtype relationships such as "every salesman is also a person". Type II corresponds to designative relationships such as 1:1, N:1 or n-ary relationships with at least one 1 cardinality. Type III appears in associative relationships such as N:N and n-ary relationships and weak entities (Date, 2000; Elmasri & Navathe, 2000). Types IV and V deserve a different analysis, as they appear as a result of late irregular alterations of logical designs (Rivero, Doorn & Ferraggine, 2004). The remaining cases correspond to *id*s, which cannot be obtained via a semantic model, but as a consequence of similar reasons as types IV and V of *rir*s (Rivero, Doorn & Ferraggine, 2001, 2004).

MAIN CHARACTERISTICS AND APPLICATIONS OF INCLUSION DEPENDENCIES

Following, there is a description of main issues about inclusion dependencies and relevant applications of this concept.

Referential Actions and Global Semantics

Update operations promote the execution of specialized triggers -the referential actions-, for the programmed maintaining of referential integrity. The actions are: *cascade, restrict, no action, set null, set default* (Date, 2000; Markowitz, 1994; SQL:1999_2, 1999; Türker & Gertz, 2001).

Despite the fact that the local effect of such rules is precisely defined, when update operations are executed on the database state, the global effects those interacting actions promote may show ambiguities (Lüdascher & May, 1998; Markowitz, 1994; May & Lüdascher, 2002; Reinert, 1996).This problem has been - and currently is - a matter of profuse research, from the beginning of the relational databases era. While in Markowitz(1994) the

Figure 1. Placements of a set of attributes in correlation with the key