Referential Constraints

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

INTRODUCTION

Inclusion dependencies support essential semantic aspects of the standard relational data model. An inclusion dependency is defined as the existence of attributes in a table whose values must be a subset of the values of the corresponding attributes in another table (Codd, 1990; Abiteboul, Hull, & Vianu, 1995; Connolly & Begg, 2004). Formally, it can be expressed as $R[X] \subseteq S[Z]$. R and S are relation names. With X and Z as compatible attributes, R[X] and S[Z] are the inclusion dependency's left and right sides respectively. When Z is the primary key of S or it is restricted by a unique clause, the inclusion dependency is key-based (also named referential integrity restriction, *rir*). In this case, X is a foreign key (FK) for R. On the contrary, if Z does not constitute the key of the relation, the inclusion dependency is non-key-based (simply, an inclusion dependency, *id*). Both *rir*s and *id*s are referential constraints.

*Rir*s are important because they contain basic local semantic characteristics, 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 et al., 1995). Conversely, *id*s do not appear as a product of the translation 'conceptual schema→logical schema', but because of ad-hoc changes made in the phase of detail design, some denormalization degree, or the presence of complex n-ary relationship constructs. In this scenario, *id*s frequently misrepresent objects and their corresponding inter-object relationships.

*Rir*s can be declaratively defined via the SQL foreign key clause (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>]

If <referenced column list> is omitted, the foreign key refers to the primary key of <referenced table name>.

The *rir*s can be specified with respect to different match types: SIMPLE (implicit if no match option is declared), PARTIAL, and FULL. As stated in the SQL:1999 standard document: If <match type> 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, FULL means that, for each row in the referencing table, either all foreign key values are null or they equal the value of the corresponding referenced column (Türker & Gertz, 2001; SQL:1999-2).

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 (a row in the right table), or the insertion of a row in the referencing (left) table. Possible actions are: *cascade*, *restrict*, *no action*, *set null*, *set default* (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. The deletion (update) of a referenced row is disallowed by *restrict* and *no action* rules, whenever at least a row in the left table is pointing to 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.

*Id*s may be defined with general CHECK statements having the semantics of assertions or triggers. Erroneously *id*s are frequently specified by means of an attribute-based CHECK constraint associated to the referencing table, requiring the existence of the referred-to value.

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

Since these constraints are checked whenever any tuple changes the value for that attribute, an update of the referred-to value in the referenced table would result in the attribute-based CHECK constraint becoming violated.

Triggers are a widespread way of implementation, although they usually complicate the development of application programs and make the integrity maintenance quite difficult (Date & Darwen, 1997; Elmasri & Navathe, 2000; Connolly & Begg, 2004).

BACKGROUND

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

Structure

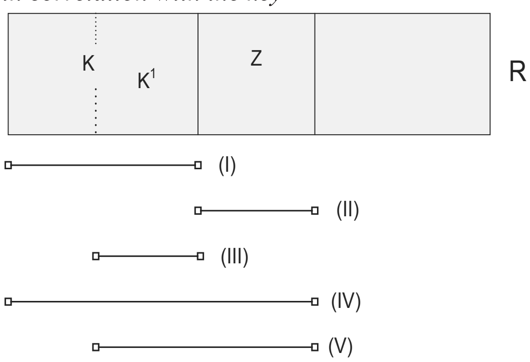
Considering a relation shape, there are five possible placements of a non-empty set of attributes with regard to the key placement. With W as such a set of attributes and K the primary key of R, the five placements are depicted in the Figure 1: (I) $W \equiv K$ (W coincides with K); (II) $W \equiv Z$, where Z is a subset of non-key attributes; (III) $W \equiv K^1$, where K_1 is a proper subset of K, $K_1 \neq \varnothing$; (IV) $W \equiv K \cup Z$; and finally (V) $W \equiv K_1 \cup Z$, $K_1 \neq \varnothing$ (W and K partially overlap). In all cases, $Z \neq \varnothing$.

As a consequence, 25 possible configurations of $R[W_R] \subseteq S[W_S]$ can be derived (see Table 1). 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 et al., 1995). Type I depicts subtype relationships; type II corresponds to designative relationships such as 1:1, N:1, or n-ary relationships with at least one 1 cardinality; and type III appears in associative relationships such as N:N, n-ary relationships, and weak entities (Elmasri & Navathe, 2000; Teorey, 1990). *Rir*s of types IV, V, and *id*s deserve a different analysis, as they cannot be derived from a conceptual model. They appear as the specification of well-typified business rules with the semantics of an inclusion in latter stages of the logical design (Rivero, Doorn, & Ferraggine, 2001, 2004).

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



K_*(key); Z_* (non-key attributes); K_*^1 (a proper subset of K_*); $_*$ = l, r (left, right)

Table 1. Possible structures for referential constraints

W_l \ W_r	I) Key (K_r)	II) Non Key (Z_r)	III) Part of a Key (K_r^1)	IV) Key + Non Key ($K_r \cup Z_r$)	V) Part of a Key + Non Key ($K_r^1 \cup Z_r$)
(I) Key (K_l)	**1.** $K_l << K_r$	**6.** $K_l \subseteq Z_r$	**11.** $K_l \subseteq K_r^1$	**16.** $K_l \subseteq K_r \cup Z_r$	**21.** $K_l \subseteq K_r^1 \cup Z_r$
(II) Non Key (Z_l)	**2.** $Z_l << K_r$	**7.** $Z_l \subseteq Z_r$	**12.** $Z_l \subseteq K_r^1$	**17.** $Z_l \subseteq K_r \cup Z_r$	**22**. $Z_l \subseteq K_r^1 \cup Z_r$
(III) Part of a Key (K_l^1)	**3.** $K_l^1 << K_r$	**8.** $K_l^1 \subseteq Z_r$	**13.** $K_l^1 \subseteq K_r^1$	**18.** $K_l^1 \subseteq K_r \cup Z_r$	**23.** $K_l^1 \subseteq K_r^1 \cup Z_r$
(IV) Key + Non Key ($K_l \cup Z_l$)	**4.** $K_l \cup Z_l << K_r$	**9.** $K_l \cup Z_l \subseteq Z_r$	**14.** $K_l \cup Z_l \subseteq K_r^1$	**19.** $K_l \cup Z_l \subseteq K_r \cup Z_r$	**14.** $K_l \cup Z_l \subseteq K_r^1 \cup Z_r$
(V) Part of a Key + Non Key ($K_l^1 \cup Z_l$)	**5.** $K_l^1 \cup Z_l << K_r$	**10.** $K_l^1 \cup Z_l \subseteq Z_r$	**15.** $K_l^1 \cup Z_l \subseteq K_r^1$	**20.** $K_l^1 \cup Z_l \subseteq K_r \cup Z_r$	**25.** $K_l^1 \cup Z_l \subseteq K_r^1 \cup Z_r$