ISO9075-2 Sec 4.44 Triggers (extract)

What constitutes a transition depends on the trigger event. If the trigger event is DELETE, a transition is a row in the old transition table. If the trigger event is INSERT, a transition is a row in the new transition table. If the trigger event is UPDATE, a transition is a row *OR* in the old transition table paired with a row *NR* in the new transition table, such that *NR* is the row derived by applying a specified update operation to *OR*. *OR* and *NR* are the *old row* and the *new row*, respectively, of the transition.

A statement-level trigger that is considered as executed for a state change *SC* (in a given statement execution context) is not subsequently executed for *SC*.

If a row-level trigger *RLT* is considered as executed for some row *R* in *SC*, then *RLT* is not subsequently executed for *R*.

A (possibly empty) old transition table exists if the trigger event is UPDATE or DELETE. It consists of a copy of each row that is to be updated in or deleted from the subject table. A (possibly empty) new transition table exists if the trigger event is UPDATE or INSERT. It consists of a copy of each row that results from updating a row in the subject table or is to be inserted into the subject table.

A <triggered action> may refer to the old transition table only if an <old transition table name> is specified for it in the <trigger definition>, and to the new transition table only if a <new transition table name> is specified for it in the <trigger definition>.

The <triggered action> of a row-level trigger may refer to a range variable ranging over the rows of the old transition table only if an <old transition variable name> is specified for it in the <trigger definition>. Similarly, the <triggered action> of a row-level trigger may refer to a range variable ranging over the rows of the new transition table only if a <new transition variable name> is specified for it in the <trigger definition>. The scope of a transition variable or transition table name is the <triggered action> of the <trigger definition> that specifies it, excluding any <SQL schema statement>s that are contained in that <triggered action>.

When a statement execution context *SEC* is created, the set of state changes *SSC* in *SEC* is empty. Let *SCj* be a state change in *SSC*. Let *TE* be the trigger event (DELETE, INSERT, or UPDATE) of *SCj* . Let *ST* be the subject table of *SCj* .

If *TE* is INSERT or DELETE, then let *PSC* be a set whose only element is the empty set.

If *TE* is UPDATE, then:

— Let *CL* be the list of columns being updated by *SSC*.

— Let *OC* be the set of column names identifying the columns in *CL*.

— Let *PSC* be the set consisting of the empty set and every subset of the set of column names of *ST* that has at least one column that is in *OC*.

Let *PSCN* be the number of elements in *PSC*. A state change *SCj*, for *j* varying from 1 (one) to *PSCN*, identified by *TE*, *ST*, and the *j*-th element in *PSC*, is added to *SSC*, provided that *SSC* does not already contain a state change corresponding to *SCj*. Transitions are added to *SCj* as specified by the General Rules of Subclause 15.7, “Effect of deleting rows from base tables”, Subclause 15.10, “Effect of inserting tables into base tables”, and Subclause 15.13, “Effect of replacing rows in base tables”.

When a state change *SCj* arises in *SSC*, one or more triggers are activated by *SCj*. A trigger *TR* is activated by *SCj* if and only if the subject table of *TR* is the subject table of *SCj*, the trigger event of *TR* is the trigger event of *SCj*, and the set of column names listed in the trigger column list of *TR* is equivalent to the set of column names listed in *SCj*.

NOTE 115 — The trigger column list is included in the descriptor of *TR*; it is empty if the trigger event is DELETE or INSERT. The trigger column list is also empty if the trigger event is UPDATE, but the <trigger event> of the <trigger definition> that defined *TR* does not specify a <trigger column list>.

For each state change *SCj* in *SEC*, the BEFORE triggers activated by *SCj* are executed before any of their triggering events take effect. When those triggering events have taken effect, any AFTER triggers activated by the state changes of *SEC* are executed.

The <triggered action> contained in a <trigger definition> for a row-level trigger can refer to columns of old transition variables and new transition variables. Such references can be specified as <column reference>s, which can be <target specificaton>s and <simple target specification>s when they refer to columns of the new transition variable in the triggered action of a BEFORE row-level trigger.

NOTE 116 — By using such <column reference>s as <asignment target>s (see [ISO9075-4]), the triggered action of a BEFORE trigger is able to cause certain SQL-data change statements to have different effects from those specified in the statements.

When an execution of the <triggered SQL statement> *TSS* of a triggered action is not successful, then an exception condition is raised and the SQL-statement that caused *TSS* to be executed has no effect on SQL-data or schemas.