PANDORA CASE TOOL: TRIGGERS GENERATING FOR CARDINALITY CONSTRAINTS IN RDBMS

Harith Al-Jumaily
Dept. of Computer Sciences, University of Carlos III in Madrid
Av. Universidad 30 - 28911 Leganés, Madrid

Dolores Cuadra
Dept. of Computer Sciences, University of Carlos III in Madrid
Av. Universidad 30 - 28911 Leganés, Madrid

Paloma Martínez
Dept. of Computer Sciences, University of Carlos III in Madrid
Av. Universidad 30 - 28911 Leganés, Madrid

ABSTRACT

The cardinality constraint is one of the most important constraints that can be established in a conceptual model. Nevertheless, not all CASE tools for developing database support the validation of this constraint due to the complexity associated to the referential integrity rules. So that, the contribution of this work concerns to provide PANDORA CASE (a tool used for learning and developing database in internet) algorithms which can be used to generate triggers for enforcing the cardinality constraints of a given database. These triggers are full compatible with the standard SQL3.

KEYWORDS

Database Development, CASE Tools, ECA rules, Cardinality constraints, and Relational Model.

1. INTRODUCTION

In any database development methodology as (Teorey, T. 1999) and (Elmasri, R. et al. 2000), there is a process devoted to transforming conceptual model into logical model. In such process, a modelling problem such as inconsistency in constraints of cardinality can be existing because the logical constructs are not coincident with conceptual constructs. In the last decade, multiple attempts to resolve the modelling problems have been developed. One of such attempts was the automation of database design process by using CASE tools. These tools try to help the database designer in different phases. Nevertheless, frequently these tools are not more than diagrams and don't completely cover all phases of database analysis and the design methodology that they propose. Therefore, it is necessary to create new tools such as PANDORA that adept more advanced focus and integrate solutions to the previous problems.

So that in this work, a proposal to providing PANDORA CASE tool triggers for controlling the cardinality constraints is presented. The problem of triggers execution (Al-Jumaily, H. et al, 2002), is solved such as in many works have been done in the area of termination (Aiken A. et al, 1995) (Baralis, E. et al, 1997). Our work has been done using static analysis and the notion of triggering graph.

The rest of this work is organised as follows. In section (2) PANDORA CASE tool is defined. Section (3) is devoted to the transformation of binary relationships trying to avoid the semantic losses. In section (4) algorithms are presented which are used to generate conceptual triggers. In section (5), some conclusions are exposed.

2. PANDORA PROJECT

As shown in Figure (1), all commercial CASE tools for design databases (e.g. Designer2000, Erwin, Power Designer...etc.) support the definition of the global database scheme, admitting graphical representations of the conceptual model, according to different database design methodologies. They save all the information about the database structure and design in a repository which is a common shared memory to design and to maintenance activities (Mokrane, B. et al. 2000). The main contribution of database design CASE tools is to convert the ER model constructs into relational model. By using the basic transformation rules. In addition, it covers different phases in a database life cycle to solve some of the modelling problems and providing processes to carry out the specific methodology.

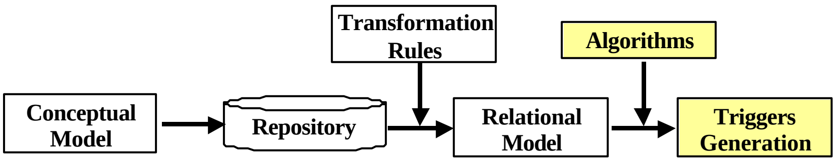

Figure 1. PANDORA CASE Tools Structure.

So, the main aim of PANDORA project is to obtain a CASE tool in internet, which solve some lacking features in several commercial CASE tools. It covers all database life cycle, especially in the requirements of analysis phase, as well the lack of methodological assistants, that is, what are the steps to be followed in database development. PANDORA platform will include a set of tools implemented using different technologies (expert systems, natural language processing, etc.) that could be used following a methodological guide. An independent use of the modules will be allowed in order to perform specific database development tasks. Our work does not cover every phase in the database life cycle, but its contribution concerns to provide PANDORA CASE tools algorithms to generate conceptual triggers, which are used to enforce the cardinality constraints. These triggers are full compatible with the standard SQL3.

3. CARDINALITY CONSTRAINT

The maximum and minimum cardinalities are defined in (Chen, P. 1976), as the maximum and minimum number of occurrences of a type entity associated to an occurrence of the other type in the relationship. Graphically, the cardinalities restrictions are represented by a label, (0,1), (1,1), (0,N), (1,N). Focusing in the transformation of conceptual model into relational model, in this work we have defined the algorithms corresponding to the required triggers that control the cardinality constraints in order to preserve their associated semantics. We have studied in detail the transformations of relationships that have not any attributes (Cuadra D. et al. 2002).

To carry out the transformation of the cardinalities of ER model into the relational model without semantic losses, mechanisms that the relational model provides us to express semantic constraints have to be used. These mechanisms are the definition of foreign keys, Delete and Update options, **`NOT NULL`** constraint, alternate keys **`UNIQUE`**, verification constraints **`CHECKS`** (Bertino, E. et al, 1997). These rules are considered in the definition of the **`FOREIGN KEY`**, and they can not maintain the cardinality constraints between the entities and relationships, so that the triggers are used (Al-Jumaily, H. et al, 2002).

In general, the database design CASE tools can support the validation of the referential integrity constraint. However, these tools can not support the validation of the cardinality constraints due to the complexity associated to the referential integrity rules. As an example, Designer2000 doesn't have the capacity to generate triggers for cardinalities constraints validation, but it provides an editor (Trigger Definition) which allows the user to define different triggers. ERwin CASE tool has only the capacity to create triggers to enforce the previous referential integrity constraint, but not to maintain the cardinality constraints. For that, designers and DBMS have the major responsibilities to protect the database semantic.

For example, let **`Ei`** and **`Ej`**, are the associated entities in binary relationship **`Rij`**, let us suppose that the foreign key options in this case are (**`On Delete Cascade`**) and (**`On Update Cascade`**). In this case the cardinality semantic losses are shown in table (1).

Table 1. Semantic losses in binary relationships N:M, 1:N and 1:1 **(I:Insert, D:Delete, U:Update)**

Type	Card(Ei)	Card(Ej)	Exist Semantic Loss, when
`N:M`	`1,N`	`0,N`	`I(Ei), D(Ej), D(Rij), U(Rij.fk_Ei)`
	`1,N`	`1,N`	`I(Ei,Ej), D(Ei,Ej), D(Rij), Rij.fk_Ei,Rij.fk_Ej)`
`1:N`	`1,N`	`1,1`	`I(Ei), D(Ej), U(Ej.fk_Ei)`
	`1,N`	`0,1`	`I(Ei), D(Ej), U(Ej.fk_Ei)`
`1:1`	`1,1`	`1,1`	`I(Ei), D(Ej), U(Ej.fk_Ei)`

4. GENERATING TRIGGERS

This work does not cover every phase in a database life cycle, but its contribution concerns to provide algorithms to generate triggers which are used to enforce the cardinality constraints of the binary relationships (N:M, 1:N):

4.1 Conceptual triggers for Entity Relations

According to section (3), semantic losses are produced when an occurrence is deleted from the entity relations. When an occurrence is deleted from the entities **`Ei`** or **`Ej,`** semantic losses are produced if the minimum cardinalities associated with (**`Ei`** or **`Ej`**)>**0,** because it is not possible to delete occurrences from **`Ei`** and to have some occurrences in **`Ej`** not connected, or vice-versa. In this case, the triggers are generated for each entity relation **`Ei`** and **`Ej`** according to the following steps:

(a) **`CREATE Trigger Del_T1_Ei,`** it is a **`Before/Row`** trigger which is activated when **`Is_It_Enable(Del_T1_Ei)`** is true. This trigger is used to disable all triggers of the related relationship **`Rij,`** and it saves the primaries keys of the deleted occurrences into global variables. The action of this trigger consists of the following SQL3 statements: -

- If the type of the binary relationship **`Rij`** is (**N:M**), and exists one of the maximum cardinalities associated with (**`Ei`** or **`Ej`**)>0, then the trigger must have the action **`Disable(gbp_Rij)`**, which disabled all triggers associated with the related relationship **`Rij`**.
- If the type of the binary relationship **`Rij`** is (**1:N**), and the maximum cardinalities associated with **`Ei`** is one (i.e. **`Ei`** is parent relation), then the trigger must have the action **`Disable(gbp_Ej)`**, which disabled all triggers associated with a child entity **`Ej`**.
- If the minimum cardinality associated with **`Ei`** is one, then the trigger must have the action **`Save(:Old.pk_Ei).`** This action is used to save the transition variable, which is used latter in the after-trigger for checking the semantic losses in a child relation **`Ei`**.

(b) **`CREATE Trigger Del_T2_Ei,`** it is an **`After/Statement`** trigger which is activated when **`Is_It_Enable(Del_T2_Ei)`** is true. This trigger is used to enable all triggers that have been disabled in the previous (**`Before/Row`**) triggers and to check the semantic losses in a relationship relation **`Rij`**. The trigger action consists of the following SQL3 statements:-

- If the type of **`Rij`** is (**N:M**), and exists one of the maximum cardinality associated with (**`Ei`** or **`Ej`**)>**0**, then the trigger action must check the semantic in **`Rij`**.
- If the type of **`Rij`** is (**1:N**), and the maximum cardinality associated with **`Ei`** is **`N`** (i.e. **`Ei`** is parent entity), then the trigger action must check the semantic in a child relation **`Ei`** after the deleting statement is actually performed.

4.2 Conceptual triggers for Relationship Relations

When an occurrence is deleted from a binary relationship **`Rij`**, semantic losses are produced when the minimum cardinalities associated with (**`Ei`** or **`Ej`**)>**0,** because it is not possible to delete from **`Rij`** and to have some occurrences in **`Ei`** or **`Ej`** not to be connected. The triggers for **`Rij`** are generated as shown:

(a) **`CREATE Trigger Del_T1_Rij`**, it is a **`Before/Row`** trigger which is activated when **`Is_It_Enable(Del_T1_Rij)`** is true. This trigger is used to save the primary key values of the deleted occurrences into global variables which are used latter in after-trigger for semantic checking. The trigger action consists of the following operations: -

- If the minimum cardinality of **`Ei`** is greater than **`0`**, then the trigger must have the action **`Save(:Old.pk_Ei)`**.
- If the minimum cardinality of **`Ej`** is greater than **`0`**, then the trigger must have the action **`Save(:Old.pk_Ej)`**.

(b) **`CREATE Trigger Del_T2_Rij`**, it is an **`After/Statement`** trigger which is activated when **`Is_It_Enable(Del_T2_Rij)`** is true. It is used to check the semantics losses in the relation **`Rij`**.

To generate the triggers that are used to enforce the cardinality constraints when the updating of an occurrence in **Rij** is actually performed. The same algorithms, which are used for controlling deletions, are used for controlling updates.

5. CONCLUSION

Generation rule (trigger) systems are exposed in many studies and some challenges will front the developers and the investigators to control the behaviours of these rules. One of these challenges is that commercial CASE tools for database developments can not cover all phases of database analysis and the real ER models. Our work presents the algorithms, which are used for triggers generation. These triggers are used to enforce the cardinality constraint of the designed database and they are fully compatible with the standard SQL3 trigger. Currently, we are performing some experiments to measure the efficiency of such triggers in relational database. For instance, access times to achieve updating operations or the increase storage space when use triggers.

In the future, we will try to integrate these algorithms in CASE tool PANDORA. In order to achieve a transformation of ER schemata into the relational model without loss of semantics. In this paper we present the preservation of cardinality constraint in binary relationships but the aim is to develop that keep all the integrity constraints or that force their verification regardless of which program accesses the databases.

REFERENCES

Aiken A. et al, 1995. Static analysis techniques for predicting the Behavior of Active Database Rules. In ACM Transactions on Database Systems, 20(1).

Al-Jumaily, H. et al, 2002. An Execution Model For Preserving Cardinality Constraints in The Relational Model. In 4th Int. Conf. On Enterprise Information Systems. Spain.

Teorey, T. 1999. Database Modeling & Design. Morgan Kaufmann Series in data management systems.

Baralis, E. et al, 1997. Better Static Rule Analysis for Active Database Systems. IBM Almaden Research Center.

Bertino, E. et al, 1997. Integrity constraint checking in Chimera, In Proc. Of the 2nd Int. Workshop on Constraints Database Systems, Delphi, Greece.

Chen, P. 1976. The Entity-Relationship Model – Toward a Unified View of Data, In ACM Transactions on Database Systems, Vol. 1, N. 1.

Cuadra D. et al. 2002. Database Integrity: Challenges and Solutions. Ed: Idea Group Publishing.

Elmasri, R. et al. 2000. Fundamentals of Database Systems. Third Edition, Addison-Wesley, 2000.

Mokrane, B. et al. 2000. CASE Tools: Computer Support for Conceptual Modeling / Advanced database technology and design. Editorial Artech House.

Paton, N. et al. 1999. Active Database Systems. In ACM Computing Surveys, Vol.31, No.1.