Automated Insertion of Exception Handling for Key and Referential Constraints

Kaiping Liu, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Hee Beng Kuan Tan, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Xu Chen, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ABSTRACT

Key and referential constraints are two main integrity constraints in database applications. These constraints can be automatically enforced by the Database Management System with their exception—violation from these constraints—handled by programmers. This paper proposes an approach to relieve the burden of programmers from mechanical coding for handling exceptions of these constraints violation by using program transformation. We first propose an extended abstract syntax tree to include SQL query semantics. Based on it, each code pattern that requires exception handling together with the exception handling code to be inserted is represented as a transformation rule. We provide two alternatives to handle the exceptions: one is to handle the exceptions in conjunction with the built-in enforcement feature in Database Management System; the other is handling them without using the feature provided in Database Management System. Hence, two types of transformation rules are provided accordingly. A tool GEHPHP (Generation of Exception Handling for PHP Systems) has been developed to implement the proposed approach. Experiments have also been conducted to evaluate the applicable of the approach.

Keywords: Constraints Violation, Database Query Aware AST (DB-AST), Exception Handling, Key and Referential Constrains, Structured Query Language (SQL) Query Pattern, Transformation Rule

INTRODUCTION

Relational database is commonly used among database applications. Key and referential constraints (KRC) are two main integrity constraints enforced in relational databases. These constraints play a key role in ensuring accuracy and consistency of data, so their enforcement during a database updating is of great importance.

Most database management systems (DBMS) can automatically enforce KRC by defining them in the database schema, upon which DBMS will reject any updating to a database that will lead to constraint violation. However, exception handling to handle such
rejections still requires coding by programmers. Following is an example which executes an SQL query but without the provision of exception handling in Box 1.

In this example, the attribute “username” is a primary key (PK). The SQL is executed without any exception handling code following. This may cause key constraint violation. In order to make the program robust and less error-prone, programmers should provide adequate exception handling of KRC violations.

Generally, there are two levels to enforce the KRC in a database application: DBMS level and application level. In the classical DBMS, only a limited range of constraints can be enforced, and referential constraints enforcement is often neglected in favor of better performance. Empirical studies indicated that DBMS-level integrity constraint enforcement is rare in actual practice (Blaha, 2004). At the application level, programmers may perform KRC by detecting any KRC violation before executing the corresponding SQL operations, though we discovered that programmers seldom perform application level KRC enforcement.

This paper proposes a novel approach to automatically generate and insert exception handling code of KRC violation. In our approach, we first propose an extended abstract syntax tree (AST) to include SQL query semantics based on traditional AST. Each code pattern that requires exception handling together with the code to be inserted as precondition or postcondition is represented as a transformation rule. Two alternatives are provided for handling the possible exceptions associated with KRC. The first handles the exceptions in conjunction with the built-in enforcement in DBMS, while the second handles the exceptions in program without using the automated enforcement feature provided in DBMS. A tool GEHPHP (Generation of Exception Handling for PHP Systems) has been developed to implement the proposed approach. Experiments have also been conducted to evaluate the applicable of the approach.

The paper is organized as follows. The next section provides the background knowledge. The section after presents the proposed approach. The section next describes the prototype system as well as the evaluation of the approach. The section after examines related work and the final section concludes.

BACKGROUND

Integrity constraints play an important role in database applications. Enforcement of these constraints will reject inconsistent or unauthorized data from entering a database. KRC are the most common integrity constraints which can be enforced automatically by a DBMS once specified in database schema (Silberschatz, Korth, & Sudarshan, 2002).

Key Constraints

A key constraint dictates an attribute or a set of attributes that identifies exclusively each record in a table. A PK is a non-null-valued attribute which is unique to each individual distinct record.

Referential Constraints

A referential constraint establishes a relationship between two tables by identifying an attribute or a set of attributes in the referencing table that refers to an attribute or a set of attributes in the referenced table. An attribute defined as

Box 1.

```php
$username = $_GET['username'];
$query = "UPDATE users SET username='$username';
mysql_query($query);
```
Interactive Query Expansion with Automatically Generated Category-Specific Thesauri
[www.igi-global.com/chapter/interactive-query-expansion-automatically-generated/30274?camid=4v1a](www.igi-global.com/chapter/interactive-query-expansion-automatically-generated/30274?camid=4v1a)

A Measurement Ontology Generalizable for Emerging Domain Applications on the Semantic Web
[www.igi-global.com/article/measurement-ontology-generalizable-emerging-domain/3365?camid=4v1a](www.igi-global.com/article/measurement-ontology-generalizable-emerging-domain/3365?camid=4v1a)