

Asserções (Assertions)

Elmasri, Ramez. Shamkant B. Navathe. "Fundamentals Of Database Systems" -- 4th ed. ISBN 0-321-12226-7.

As restrições de integridade fornecem meios para assegurar que mudanças feitas no banco de dados por usuários autorizados não resultem na inconsistência dos dados. Há vários tipos de restrições de integridade que seriam cabíveis a bancos de dados. Entretanto as regras de integridade são limitadas às que podem ser verificadas com um mínimo de tempo de processamento.

Uma asserção (afirmação) é um predicado expressando uma condição que desejamos que o banco de dados sempre satisfaça. Restrições de domínio, dependências funcionais e restrições de integridade referencial são formas especiais de asserções. Quando uma assertiva é criada, o sistema testa sua validade. Se a afirmação é válida, então qualquer modificação posterior no banco de dados será permitida apenas quando a asserção não for violada.

O alto custo de testar e manter asserções tem levado a maioria de desenvolvedores de sistemas a omitir o suporte para asserções gerais. Em SQL, os usuários podem especificar as restrições genéricas via asserções declarativas, usando a declaração CREATE ASSERTION da DDL. Em uma asserção, é dado um nome à restrição por meio de uma condição semelhante à cláusula WHERE de uma consulta SQL. Por exemplo, para especificar a restrição "O salário de um empregado não pode ser maior que o salário do gerente do departamento em que ele trabalha" em SQL, precisará formular a seguinte asserção:

```
CREATE ASSERTION LIMITE_SALARIO
CHECK (NOT EXISTS
  (SELECT *
   FROM EMPREGADO E, EMPREGADO M, DEPARTAMENTO D
   WHERE E.SALARIO>M.SALARIO AND
        E.DNO=D.NUMERO AND
        D.SSNGER=M.SSN) );
```

A restrição LIMITE_SALARIO é seguida pela palavra-chave CHECK, que é seguida por uma condição entre parênteses que precisa ser verdadeira em todos os estados do banco de dados para que a asserção seja satisfeita. O nome da restrição pode ser usado posteriormente para sua referência, modificação ou eliminação. O SGBD é responsável por garantir que a condição não seja violada. Pode ser usada qualquer condição na cláusula WHERE, mas muitas condições podem ser especificadas utilizando-se estilo EXISTS e NOT EXISTS das condições SQL. Sempre que alguma tupla no banco de dados fizer com que uma condição ASSERTION evolua para FALSE, a restrição é violada. A restrição será satisfeita para um dado estado no banco de dados, se nenhuma combinação de tuplas no banco de dados violar essa restrição.

A técnica básica para a formulação de asserções é especificar uma consulta que selecione as tuplas que violem a condição desejada. Por meio da inclusão dessa consulta em uma cláusula NOT EXISTS, a asserção especificará que o resultado dessa consulta deverá ser vazio. Logo, a asserção será violada se o resultado da consulta não for vazio. Em nosso exemplo, a consulta seleciona todos os empregados cujos salários sejam maiores do que o do gerente de seu departamento. Se o resultado dessa consulta não for vazio, a asserção será violada.

Observe que a cláusula CHECK e a condição da restrição podem ser usadas também para especificar restrições nos atributos e nos domínios e nas tuplas. A principal diferença entre uma CREATE ASSERTION e as outras duas é que a cláusula CHECK da SQL em um atributo, um domínio ou uma tupla fará a checagem somente quando as tuplas forem inseridas ou atualizadas. Assim, a verificação poderá ser implementada com maior eficiência pelo SGBD nesses casos. O projetista do esquema poderia usar CHECK em atributos, nos domínios e nas tuplas apenas quando estiver certo de que a restrição só poderá ser violada pela inserção ou atualização das tuplas. Porém, o projetista deveria usar CREATE ASSERTION somente nos casos em que não for possível usar CHECK nos atributos, nos domínios ou nas tuplas, assim a verificação será implementada com maior eficiência pelo SGBD.

Exemplos

1) Definir uma restrição de integridade que não permita saldos negativos.

```
CREATE ASSERTION SALDO_RESTRICAO1
CHECK (NOT EXISTS
(SELECT *
FROM CONTA
WHERE SALDO < 0) )
```

2) Só permitir a inserção de saldos maiores que quantias emprestadas para aquele cliente.

```
CREATE ASSERTION SALDO_RESTRICAO2
CHECK (NOT EXISTS
(SELECT *
FROM CONTA
WHERE SALDO <
(SELECT MAX(quantia)
FROM EMPRESTIMO
WHERE EMPRESTIMO.CLIENTE_COD = CONTA.CLIENTE_COD) ) )
```