Banco de Dados

Restrições de Integridade, Asserções e Triggers

Criando e Gerenciando Usuários

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

Banco de Dados

Triggers em SQL

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

7. Restrições de Integridade

- Triggers -

- Primeiras gerações de SGBDs
  - Armazenavam dados de forma passiva
    - Nenhuma ação era executada pelo SGBD
  - Ações sobre dados só eram executadas quando especificadas por transações
  - Muitas restrições de integridade deveriam ser mantidas por aplicativos
    - **Total de salários por departamento deve ser menor que K**
      - Ao se atualizar salários
        - Garantir a restrição de total de salários por departamento
      - Para cada novo valor K'
        - Alterar todos os programas que atualizam salários
    - **Restrições que envolvam mais de uma tabela**
      - Um estudante não pode ser um empregado com nível superior completo

7. Restrições de Integridade

- Triggers -

- Regras ativas
  - Mecanismos que especificam ações a serem executadas quando certas condições são satisfeitas
  - Definidas inicialmente para **sistemas de bancos de dados ativos**
    - Armazenam dados e regras
    - Disparam e executam as ações definidas nas regras
- Regras E[C]\A

  Na ocorrência de um EVENTO

  Se CONDIÇÃO

  Então execute AÇÃO
  - Quando um evento ocorre
    - Uma ou mais ações podem ser disparadas, caso
      - A condição seja satisfeita
  - Um evento pode representar uma atualização sobre dados do BD

7. Restrições de Integridade

- Triggers -

- *Triggers*
  - Mecanismos de regras ativas implementados por SGBDs Relacionais existentes no mercado
  - São armazenados no banco de dados
  - Eventos
    - Insert, Delete, Update
  - Execução das ações
    - Antes ou depois da ocorrência do evento (MySQL)
      - SQL Server
        - Depois
    - A condição de execução seja satisfeita
      - Definida dentro de um *trigger*
  - Definidos através de uma expressão DLL
    - *Create trigger*

Banco de Dados

Triggers no MySQL

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

7. Restrições de Integridade

- Triggers -

- Sintaxe da cláusula CREATE TRIGGER (MySQL)

```
CREATE
    [DEFINER = { user | CURRENT_USER }]
    TRIGGER nome_trigger
    trigger_time trigger_event
    ON nome_tabela FOR EACH ROW
    [trigger_order]
    trigger_body

trigger_time: { BEFORE | AFTER }

trigger_event: { INSERT | UPDATE | DELETE }

trigger_order: { FOLLOWS | PRECEDES } outro_nome_trigger
```

7. Restrições de Integridade

- *Triggers* -

- Sintaxe da cláusula CREATE TRIGGER (MySQL)

trigger_body:

- É a sentença que será executada quando o gatilho for ativado;
- Para executar múltiplas sentenças, usar os delimitadores de bloco BEGIN … END;
- Dentro do corpo da trigger, você pode se referir a colunas na tabela atual (a tabela associada com a trigger) usando os aliases OLD e NEW.
- OLD.col_name referencia ao valor de uma linha existente antes da mesma ser atualizada ou deletada;
- NEW.col_name referencia ao valor de uma nova linha a ser inserida ou uma linha existente depois da mesma ser atualizada;
- Triggers não podem usar NEW.col_name ou OLD.col_name para se referir a colunas calculadas (geradas).

7. Restrições de Integridade

- Triggers -

- Cláusula CREATE TRIGGER (MySQL)
  - Observações:
    - Triggers só podem ser associadas com tabelas base, nunca com tabelas temporárias ou views (visões);
    - Ações de cascateamento de chaves estrangeiras NÃO ativam triggers;

7. Restrições de Integridade

- Triggers -

- Exemplo1 (MySQL):

  - Definir um gatilho de tal maneira que sempre que se inserir um registro (uma tupla) na tabela “TABLE_1”, sejam armazenadas a hora atual na coluna (atributo) “MY_DATETIME_COLUMN” e a data atual na coluna “MY_DATE_COLUMN”:

```
CREATE TRIGGER mytrigger
BEFORE INSERT ON TABLE_1
FOR EACH ROW
BEGIN
SET NEW.MY_DATETIME_COLUMN = NOW(),
NEW.MY_DATE_COLUMN = CURDATE()
END;
```

7. Restrições de Integridade

- Triggers -

- Exemplo2 (MySQL):

Definir um gatilho de tal maneira que o menor salário possível a ser associado a um empregado (funcionário) seja o valor de R$1000,00

...

```
CREATE TRIGGER Sal_minimo
BEFORE INSERT ON `Empregado`
FOR EACH ROW
BEGIN
    IF NEW.Salario IS NOT NULL AND NEW.Salario<1320.00 THEN
        SET NEW.Salario=1320.00;
    END IF;
END;//
```

...

7. Restrições de Integridade

- Triggers -

- Uso de BEGIN e END; no <trigger_body> (MySQL):
  - Como podem ser utilizados vários comandos, com o delimitador de comandos: “;” , deve-se redefinir o delimitador do MySQL momentaneamente, usando, por exemplo, o comando:
  - delimiter // – para alterar o delimitador para “//”, então teremos:
  - CREATE TRIGGER Sal_minimo

```
BEFORE INSERT ON `Empregado`
FOR EACH ROW
BEGIN
     IF NEW.Salario IS NOT NULL AND NEW.Salario<1320.00 THEN
        SET NEW.Salario=1320.00;
    END IF;
END;//
```

  - E “delimiter ;”, com o retorno ao delimitador “padrão”

7. Restrições de Integridade

- Triggers -

- Exemplo3 (MySQL):
  - Garantir que o saldo da conta fique entre 0 e 100 (0<=saldo<=100):

```
delimiter //
CREATE TRIGGER upd_check
BEFORE UPDATE ON Conta
FOR EACH ROW
BEGIN
    IF NEW.saldo < 0 THEN
        SET NEW.saldo = 0;
    ELSEIF NEW.saldo > 100 THEN
        SET NEW.saldo = 100;
    END IF;
END;//
delimiter ;
```

7. Restrições de Integridade

- Triggers -

- Exercícios
  - Especificar as seguintes restrições de integridade no MySQL
    - O volume de salários pagos em um departamento não pode ser maior que 15000
    - Salário de um empregado não pode ser maior que o salário do chefe de departamento

PS: Dado que o esquema do banco de dados 'Empresa' foi modificado para o modelo mostrado abaixo:

7. Restrições de Integridade

- Triggers -

- Exercícios
  - ⇨O volume de salários pagos em um departamento não pode ser maior que 15000

```
-- Trigger para tratar o Insert (Inserção) sobre a tabela Departamento
DELIMITER $$

USE `FuncDepart`$$
DROP TRIGGER IF EXISTS `tr_depart` $$
USE `FuncDepart`$$
CREATE TRIGGER `tr_depart` BEFORE INSERT ON `Departamento` FOR EACH ROW
BEGIN
      IF NEW.Sal_total > 15000 THEN
            SET NEW.Sal_total = 15000;
      END IF;
END$$

DELIMITER ;
```

7. Restrições de Integridade

- Triggers -

- Exercícios
  - ⇨O volume de salários pagos em um departamento não pode ser maior que 15000

```
-- Trigger para tratar o Update (Atualização) sobre a tabela Departamento
DELIMITER $$

USE `FuncDepart`$$
DROP TRIGGER IF EXISTS `Departamento_BEFORE_UPDATE` $$

CREATE DEFINER = CURRENT_USER TRIGGER `Departamento_BEFORE_UPDATE`
BEFORE UPDATE ON `Departamento` FOR EACH ROW
BEGIN
      IF NEW.Sal_total > 15000 THEN
            SET NEW.Sal_total = 15000;
      END IF;
END$$

DELIMITER ;
```

7. Restrições de Integridade

- Triggers -

- Exercícios
  - Salário de um empregado não pode ser maior que o salário do chefe de departamento

```
DELIMITER $$

USE `FuncDepart`$$
DROP TRIGGER IF EXISTS `Funcionario_AFTER_INSERT` $$

CREATE DEFINER = CURRENT_USER TRIGGER `Funcionario_AFTER_INSERT` AFTER
INSERT ON `Funcionario` FOR EACH ROW
BEGIN
      IF NEW.Salario > (SELECT G.Salario FROM (Funcionario F JOIN Departamento D ON
NEW.CDep = D.Codigo) JOIN Funcionario G ON D.CPF_Chefe = G.CPF WHERE F.CPF =
NEW.CPF)
        THEN
            DELETE FROM `Funcionario` WHERE CPF = NEW.CPF;
      END IF;
END;$$

DELIMITER ;
```

7. Restrições de Integridade

- Triggers -

- Exemplo:
  - Garantir que, na ocorrência de saldo devedor (saldo negativo) na relação “conta”, deve ser criado um empréstimo, de tal forma que:
    - O número da conta do depositante será o mesmo número da conta do *tomador* do empréstimo;
    - Será criado um novo empréstimo (na tabela *emprestimo*) com mesmo número da conta e nome agência, mas com saldo de empréstimo positivo (sendo o módulo do saldo devedor, ou saldo que seria negativado);
    - Dessa forma, o saldo da *conta* deve permanecer zerado (igual a 0).

7. Restrições de Integridade

- *Triggers* -

- Cláusula CREATE TRIGGER (Exemplo em SQL:1999)

```
create trigger sld_devedor after update of saldo on conta
referencing new row as nrow
for each row
when nrow.saldo < 0
begin atomic
   insert into tomador
    (select nome_cliente, numero_conta
    from depositante
    where nrow.numero_conta=depositante.numero_conta);
   insert into emprestimo values
    (nrow.numero_conta, nrow.nome_agencia, -nrow.saldo);
   update conta set saldo = 0
    where conta.numero_conta = nrow.numero_conta
end
```

7. Restrições de Integridade

- *Triggers* -

- Cláusula CREATE TRIGGER (Exemplo em SQL:1999)

```
create trigger sld_devedor after update on conta
referencing new row as nrow
for each row
when nrow.saldo < 0
begin atomic
   insert into emprestimo values
    (nrow.numero_conta, nrow.nome_agencia, -nrow.saldo);
   update conta set saldo = 0
    where conta.numero_conta = nrow.numero_conta
end
```