Triggers

A tradução de trigger é *gatilho*. É uma condição que dispara a execução de trechos de códigos armazenados no servidor, sem que haja uma chamada específica a eles. Tais trechos são executados a partir de uma situação satisfeita no servidor ou então em situações temporais em horários programados. Este segundo tipo também é conhecido como *event schedulers*.

Esta idéia permite automatizar certas operações quando alguma condição ocorrer no banco de dados. Por exemplo, quando o estoque de um produto baixa além de um limite, ou quando um orçamento ultrapassa certo valor, entre outros. A alternativa ao uso de triggers seria programar esta verificação dentro do programa construído, mas haveria pelo menos 2 problemas: a) construir o código que faz a verificação e b) garantir que ele seja chamado periodicamente para fazer a verificação. Usando triggers, há uma economia, já que a verificação só é feita quando ocorre mudança no banco de dados.

Conceitualmente o trigger pode ser disparado antes (BEFORE) ou depois (AFTER) de uma operação no banco de dados. Estas duas possibilidades podem ser chamadas na operação de inserção (INSERT), atualização (UPDATE) ou exclusão (operação DELETE).

Para criar um trigger, o comando é

```
DELIMITER $$
CREATE TRIGGER <nome>
   <tipo> ON <tabela> FOR EACH ROW
   BEGIN
      ...código SQL ...
   END$$
DELIMITER ;
```

O parâmetro tipo pode ser: BEFORE INSERT, AFTER INSERT, BEFORE UPDATE, AFTER UPDATE, BEFORE DELETE ou AFTER DELETE. O parâmetro `<tabela>` é o nome da tabela que sofrerá ação do trigger. O banco de dados onde o trigger é definido e usado deve ser tornado ativo por meio de um comando `USE`. O trigger pode e deve ser definido em um arquivo txt com a extensão `.sql` e depois pode ser ativado usando o comando SOURCE.

Veja o exemplo:

```
CREATE TABLE contabilidade (codcon INT, valor
DECIMAL(10,2));
Query OK, 0 rows affected (0.03 sec)
CREATE TRIGGER soma1 BEFORE INSERT ON
  contabilidade FOR EACH ROW SET @soma = @soma
  + NEW.valor;
Query OK, 0 rows affected (0.06 sec)
...
SET @soma = 0;
INSERT INTO contabilidade VALUES(137,14.98),
   (141,1937.50),(87,-100.00);
SELECT @soma AS 'Total incluido';
+------------------+
|  Total incluido  |
+------------------+
|  1852.48         |
+------------------+
```

Para alterar uma trigger é necessário excluí-la e depois de corriga incluí-la novamente. O comando para excluir uma trigger é

```
DROP TRIGGER <nome>
```

As palavras `NEW` e `OLD` acessam os campos antes e depois da alteração. (em INSERT não há OLD, em DELETE não há NEW. Em UPDATE há ambas. A coluna OLD é apenas de leitura, mas a NEW pode ser atualizada pelo trigger. Em um trigger de BEFORE, o valor NEW para uma coluna auto-incrementada é `0` e não o valor real que ela terá após a inclusão. Dentro da construção `BEGIN` e `END` múltiplos comandos SQL podem ser colocados. Nesse caso também é necessário readequar o delimitador. Veja-se o exemplo

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

É fácil definir uma stored procedure separadamente e chamá-la no trigger via comando `CALL`. Isto é vantajosos se o mesmo código deve ser executado em diversos triggers.

Finalmente, o trigger de uma tabela pode sem nenhuma dificuldade fazer referências a outras tabelas, desde que propriamente citadas no código.

Para poder criar triggers de tempo (event schedulers) é necessário garantir que exista um processo ativo para isso no mysql. A maneira de verificar é emitindo o comando `SHOW PROCESSLIST;`. Se na lista ele aparecer como `suspended`, é necessário ativá-lo com o comando `SET GLOBAL event_scheduler = ON;`. Uma vez ativado, seu valor na processlist deve ser `sleeping`. Para desabilitar o event scheduler deve-se atribuir a ele o valor OFF. Para o gerenciamento do event scheduler é importante a visualização dos eventos cadastrados no MySQL. O comando que mostra isto é `SHOW EVENTS;`. Para ter acesso às informações de um determinado evento, o comando é `SHOW CREATE EVENT <nome>`. O comando para criar um evento é

```
CREATE  [DEFINER = { user | CURRENT_USER }]
EVENT  [IF NOT EXISTS]   event_name
ON SCHEDULE schedule
[ON COMPLETION [NOT] PRESERVE]
[ENABLE | DISABLE | DISABLE ON SLAVE]
[COMMENT 'comment'] DO event_body;
schedule:
AT timestamp [+ INTERVAL interval] ...
| EVERY interval [STARTS timestamp
[+ INTERVAL interval] ...]
[ENDS timestamp [+ INTERVAL interval] ...]
interval:
quantity {YEAR | QUARTER | MONTH | DAY | HOUR |
MINUTE | WEEK | SECOND | YEAR_MONTH | DAY_HOUR |
DAY_MINUTE | DAY_SECOND | HOUR_MINUTE |
HOUR_SECOND | MINUTE_SECOND}
```

O parâmetro `<tempo>` é uma data e hora no formato `'yyyy-mm-dd hh:mm:ss'`. O parâmetro `<interv>` é um valor qualquer seguido de uma das unidades: `YEAR`, `QUARTER`, `MONTH`, `DAY`, `HOUR`, `MINUTE` `YEAR_MONTH`, `DAY_HOUR`, `DAY_MINUTE`, `DAY_SECOND`, `HOUR_MINUTE`, `HOUR_SECOND` ou `MINUTE_SECOND`.

Os requisitos mínimos para um CREATE EVENT válido são: as palavras CREATE EVENT seguido por um nome que o identifica. A cláusula ON SCHEDULE que determina quando e quão frequentemente ele roda e a cláusula DO que contém o(s) comando(s) SQL a executar. Eis um exemplo mínimo

```
CREATE EVENT primeiro
   ON SCHEDULE AT CURRENT_TIMESTAMP
     + INTERVAL 1 HOUR
   DO UPDATE tabela22 SET coluna = coluna + 1;
```

Este comando cria um evento chamado primeiro que roda uma única vez 1 hora após a sua criação e que incrementa a coluna citada.

Outra possibilidade seria estabelecer a execução em algum ponto no futuro, como em

```
ON SCHEDULE AT '2021-12-10 23:59:00'
```

Com alguma unidades de tempo podem-se usar unidades complexas como em dois mminutos e dez segundos.

```
+ INTERVAL '2:10' MINUTE_SECOND.
```

Duas ou mais unidades diferentes podem ser somadas, desde que cada um seja acompanhada da cláusula + INTERVAL.

Para repetir ações em intervalos regulares, deve-se usar a cláusula EVERY que deve ser seguida por um intervalo como descrito acima (só que sem o +, este não é permitido aqui). A cláusula pode conter um STARTS seguido de um timestamp indicando quando começa o ciclo de repetição. Igualmente para ENDS. Se a ação não termina dentro do intervalo de tempo, pode-se ter vários eventos iguais executando simultaneamente. Normalmente, depois que um evento é expirado ele é imediatamente eliminado. Este comportamento pode ser evitado escrevendo ON COMPLETION PRESERVE. Ao usar ON COMPLETION NOT PRESERVE apenas se deixa explícito o comportamento da coisa.

Mais alguns exemplos:

```
CREATE EVENT e_hourly
ON SCHEDULE
EVERY 1 HOUR
COMMENT 'Clears out sessions table each hour.'
DO
DELETE FROM site_activity.sessions;

delimiter |
CREATE EVENT e_daily
ON SCHEDULE
EVERY 1 DAY
COMMENT 'Saves total number of sessions
   then clears the table each day'
DO
BEGIN
INSERT INTO site_activity.totals (time, total)
SELECT CURRENT_TIMESTAMP, COUNT(*)
FROM site_activity.sessions;
DELETE FROM site_activity.sessions;
END |
delimiter ;

delimiter |
CREATE EVENT e
ON SCHEDULE
EVERY 5 SECOND
DO
BEGIN
DECLARE v INTEGER;
DECLARE CONTINUE HANDLER FOR
    SQLEXCEPTION BEGIN END;
SET v = 0;
WHILE v < 5 DO
INSERT INTO t1 VALUES (0);
UPDATE t2 SET s1 = s1 + 1;
SET v = v + 1;
END WHILE;
END |
delimiter ;

CREATE EVENT e_call_myproc
ON SCHEDULE
AT CURRENT_TIMESTAMP + INTERVAL 1 DAY
DO CALL myproc(5, 27);
```

☞ Para você fazer

Examine o arquivo D93-001.myd e crie o banco de dados associado a ele. Use como definição de tabela, o seguinte:

```
Tabela: VIAG - lista de 1500 viagens realizadas
VIAGCODV, numérico, 4: codigo da viagem
VIAGCODL, numérico, 4: código da linha
VIAGDATA, cadeia, 12: data, formato 'YYYY-MM-DD'
VIAGPASS, numerico,2: número de passageiros
```

Depois, crie vários triggers para esta tabela, a saber:

1. Obter em uma variável a soma dos passageiros transportados para códigos de linha intermunicipal começando com 7 .
2. Obter em uma variável a soma do número de passageiros ocorrido em alguma viagem feita no mês de 09 de todos os anos.
3. Anexe a esta folha cópia dos dois triggers pedidos.

Responda a seguir os 2 valores pedidos

soma passag	soma passag