Database Technology

Topic 6: Triggers and Stored Procedures

Olaf Hartig

olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Triggers

What are Triggers?

- Specify actions to be performed by the DBMS when certain events and conditions occur
- Used to monitor the DB and enforce business rules
  - Raise an alarm (e.g., constraint violation)
  - Enforce a constraint (e.g., by updating related data)
  - Update derived data in (possibly some other) table
- Typically, triggers consist of three components:
  - *Event:* update operations that activate the trigger
  - *Condition:* determines if action should be executed
  - *Action:* specifies what to do (e.g., execute stored procedure, perform sequence of SQL statements)

Example

- “Salaries cannot be increased by more than 10%.” The following trigger enforces this 10%-increase limit.

```
CREATE TRIGGER LimitSalaryTrigger
BEFORE UPDATE ON Employee
FOR EACH ROW
WHEN ( NEW.Salary > 1.1 * OLD.Salary )
SET NEW.Salary = 1.1 * OLD.Salary;
```

Using Triggers

- **CREATE TRIGGER** <name>
  { **BEFORE** | **AFTER** } <event>
  ON <table name>
  FOR EACH ROW
  [**WHEN** <condition>]
  < trigger statement(s) >**;**

[**INSERT** | **UPDATE** | **DELETE**]

Must be permanent table
(not a view or a temporary table)

Use **OLD.**<attr.name> to refer to an attribute of a row before the event

Use **NEW.**<attr.name> to refer to an attribute of a row after the event

- **SHOW TRIGGERS;**
- **DROP TRIGGER** <trigger name>**;**

BEFORE versus AFTER

- BEFORE trigger activated by attempt to insert or to modify the row, regardless of whether the attempt subsequently succeeds
- AFTER trigger activated only if the BEFORE trigger (if any) and the row operation both execute successfully
- If error during either a BEFORE or an AFTER trigger, the entire statement that activated the trigger fails

Stored Procedures

Stored Procedures – What and Why

- What are stored procedures?
  - Program modules stored in the DBMS
  - May be written in a general-purpose programming language
  - Alternatively, made of SQL commands (e.g., queries, update statements)

- Why is this useful?
  - Reduces duplication of effort if a database program is needed by several applications
  - Reduce data transfer and communication cost (assuming a client-server setting)
  - Can be used to check for complex constraints

Using Stored Procedures in SQL

- **CREATE PROCEDURE** <proc. name> **(** <params> **)**
  <local declarations>
  <procedure body>**;**

[**IN** | **OUT** | **INOUT**] <param. name> <type>

- **CALL** <proc. name> **(** <argument list> **);**
- **DROP PROCEDURE** [**IF EXISTS**] <proc. Name>**;**

- **CREATE FUNCTION** <function name> **(** <params> **)**
  RETURNS <return type>
  <local declarations>
  <procedure body>**;**

Must contain
RETURN …;

Example

```
mysql> delimiter //
mysql> CREATE PROCEDURE showsalary(IN eid INT)
    -> BEGIN
    -> SELECT salary FROM emp WHERE id=eid;
    -> END;
    -> //
mysql> delimiter ;
mysql> CALL showsalary(1);
+--------+
| salary |
+--------+
|  10000 |
+--------+
```

Another Example

```
mysql> delimiter //
mysql> CREATE PROCEDURE myproc(OUT param1 INT)
    -> BEGIN
    -> SELECT COUNT(*) INTO param1 FROM t;
    -> END;//
mysql> delimiter ;
mysql> CALL myproc(@a);
mysql> SELECT @a;
+------+
| @a   |
+------+
| 3    |
+------+
```

!!!

SQL / Persistent Stored Modules

- SQL/PSM: a set of extensions to SQL
  - General-purpose programming constructs in SQL
  - Can be used to write stored procedures
- Lots of features
  - Conditional branching
    - `IF … THEN … [ELSE …] END IF;`
    - `CASE … WHEN … THEN … [ … ] END CASE;`
  - Looping
    - `WHILE … DO … END WHILE;`
    - `REPEAT … UNTIL … END REPEAT;`
  - etc.

SQM/PSM Example

!!!

```
CREATE FUNCTION dept_size( dno INT )
 RETURNS VARCHAR(7)

BEGIN

# number of employees
DECLARE n INT;

SELECT COUNT(*) INTO n FROM emp WHERE Dept=dno;

IF n > 25 THEN RETURN "large"
ELSEIF n > 10 THEN RETURN "medium"
ELSE RETURN "small"
END IF;

END;
//
```

Summary

Summary

- *Triggers:* specify actions to be performed by DBMS when certain events and conditions occur
  - Used to monitor the DB, enforce business rules
  - Consist of event, condition, and action
- *Stored procedures:* program modules stored in DBMS
  - SQL commands
  - General-purpose programming constructs

www.liu.se

LINKÖPING UNIVERSITY