IT360: Applied Database Systems

Stored Procedures and Triggers
(Chapter 7, 10b – Kroenke,
Chapter 13 p 316 - PHP and MySQL
Web Development)

Today

- Stored procedures
- Triggers
Stored Procedures

- A stored procedure is a program that is stored within the database and is compiled when used.

- Stored procedures can receive input parameters and they can return results.

- Stored procedures can be called from:
  - Standard languages
  - Scripting languages
  - SQL command prompt

Stored Procedure Advantages

- Greater security as store procedures are always stored on the database server.
- SQL can be optimized by the DBMS compiler.
- Code sharing resulting in:
  - Less work
  - Standardized processing
  - Specialization among developers
Create And Execute Stored Procedures

- CREATE PROCEDURE proc_name (param_list)
  proc_code

- call proc_name(value1, …)

Stored Procedure Example

- Students (SNb, SName, Email, Gender, ClassYear, GPA)

- Procedure: Insert a student only if ClassYear < 2015
CREATE PROCEDURE insertStudents (SNbvar int, SNamevar varchar(50), Emailvar varchar(100), Gendervar char(1), ClassYearvar int, GPA double)
BEGIN
IF ClassYearvar < 2015 THEN
   INSERT INTO Students(SNb, SName, Email, Gender, ClassYear, GPA)
   VALUES (SNbvar, SNamevar, Emailvar, Gendervar,ClassYearvar, GPA);
END IF;
END $$
DELIMITER ;

To run: call insertStudents(7, 'John', 'john@usna.edu', 'm', 2011, null)

Class Exercise

- Add code to the previous procedure to prevent anyone named ‘Doe, John’ to be inserted into the DB.

- Test the procedure (call ....)
Triggers

- **Trigger**: stored program that is executed by the DBMS whenever a specified event occurs
- Associated with a table [or view]
- Two [or three] trigger types: BEFORE, AFTER, [and INSTEAD OF]
- Each type can be declared for INSERT, UPDATE, or DELETE

Uses for Triggers

- Provide complex default values
- Enforce data constraints
- Update views – not in MySQL
- Perform referential integrity actions
Create Trigger – MySQL Syntax

CREATE TRIGGER trigger_name
  trigger_time trigger_event
  ON table_name
  FOR EACH ROW trigger_code

trigger_time: BEFORE, AFTER
trigger_event: INSERT, DELETE, UPDATE

Trigger for Enforcing a Data Constraint – MySQL

DELIMITER $$ /* change default delimiter */
CREATE TRIGGER UpdateGPA /* provide trigger name */
AFTER UPDATE ON Enrolled /* specify when trigger should be invoked */
FOR EACH ROW
BEGIN
  DECLARE varNewGPA double;
  IF (new.Grade is not null) THEN
    /* create a variable @newGPA to compute and store the new gpa value */
    set varNewGPA = (select avg(PointGrade) from Enrolled E, Grade G
      where E.Grade = G.LetterGrade and E.SNb = new.SNb);
    /* update the Students table with the new GPA value */
    update Students set GPA = varNewGPA where SNb = new.SNb;
  END IF;
END; $$
DELIMITER ; /* restore the default delimiter */
Class Exercise

- Students (SNb, SName, Email, Gender, ClassYear, GPA)
- Define a trigger: if inserted email is null, change the Email to mxxx@usna.edu, where xxx is the SNb.
  - Hint: in MySQL, CONCAT(a,b) will return the concatenate the two string a and b
- Insert rows to test the trigger

Triggers vs. Stored Procedures

- **Trigger**
  - Module of code that is called by the DBMS when INSERT, UPDATE, or DELETE commands are issued
  - Assigned to a table or view
  - Depending on the DBMS, may have more than one trigger per table or view
  - Triggers may issue INSERT, UPDATE, and DELETE commands and thereby may cause the invocation of other triggers
- **Stored Procedure**
  - Module of code that is called by a user or database administrator
  - Assigned to a database, but not to a table or a view
  - Can issue INSERT, UPDATE, and DELETE commands
  - Used for repetitive administration tasks or as part of an application