Triggers and Stored Procedures

(Chapter 7, 11: Kroenke)

Today

- Triggers
- Stored procedures

Triggers

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

Uses for Triggers

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

Create Trigger – Generic Syntax

- CREATE TRIGGER *trigger_name*

  ON *table_or_view_name*

  AFTER | BEFORE | INSTEAD OF

  INSERT | UPDATE | DELETE

  AS

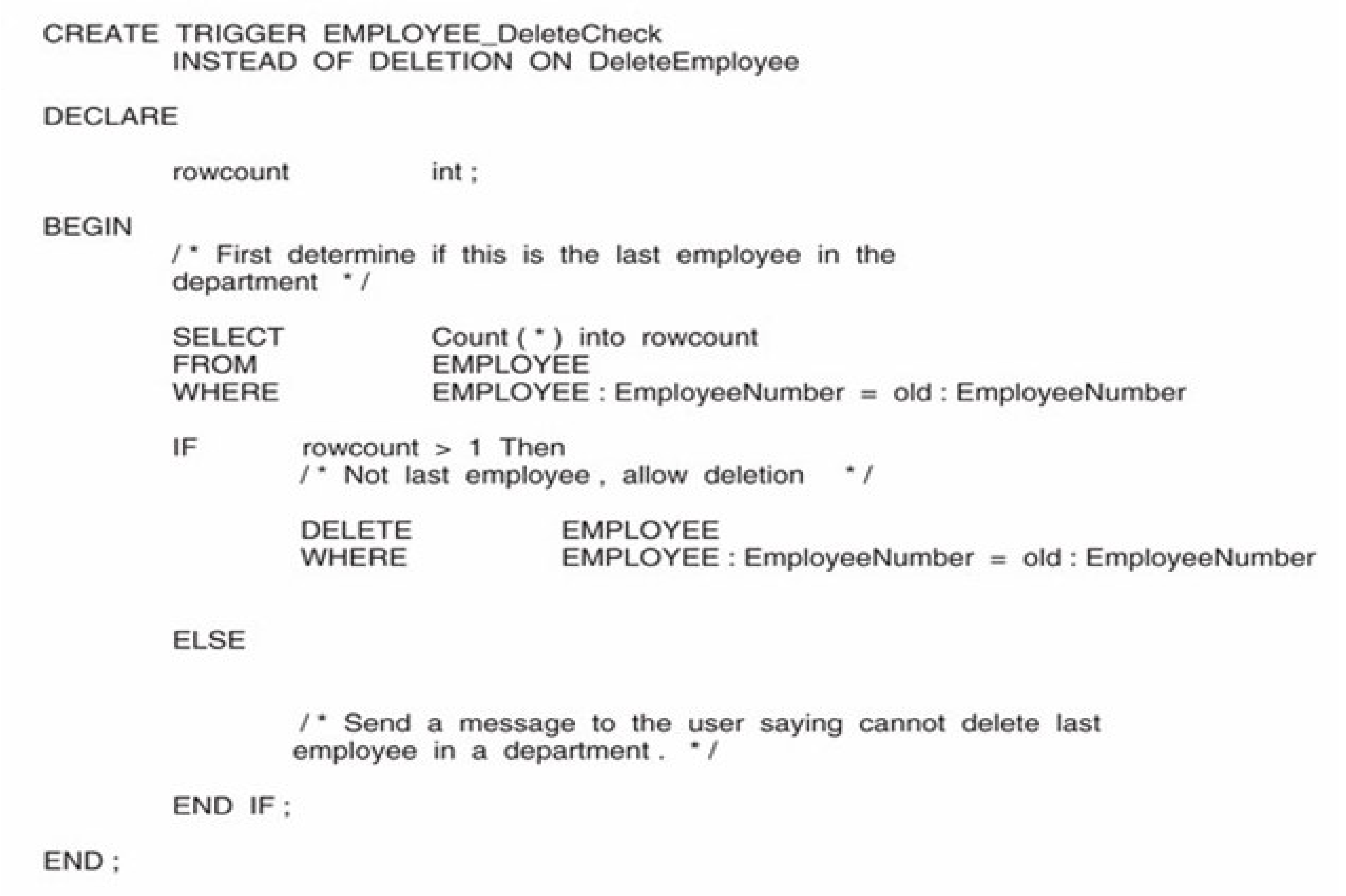
  trigger_code

Trigger for Enforcing a Data Constraint – SQL Server

Arenas (ArenaID, ArenaName, City, ArenaCapacity), ArenaCapacity >= 5000

```
CREATE TRIGGER minseating ON Arenas       /*trigger associated to Arenas*/
FOR INSERT                                /*executed after an insert*/
AS
DECLARE @capacity as int                         /*variable declarations */

SELECT @capacity = ArenaCapacity                 /* get values inserted */
FROM inserted

if @capacity < 5000

BEGIN
   ROLLBACK                                          /*undo the insert*/
   Print 'Arena too small‘                      /*message for the user*/
END
```

Trigger for Referential Integrity Actions – **pseudo-code**



```
CREATE TRIGGER EMPLOYEE_DeleteCheck
          INSTEAD OF DELETION ON DeleteEmployee

DECLARE

          rowcount            int ;

BEGIN
          /* First determine if this is the last employee in the
          department   */

          SELECT              Count ( * ) into rowcount
          FROM                EMPLOYEE
          WHERE               EMPLOYEE : EmployeeNumber = old : EmployeeNumber

          IF        rowcount > 1 Then
                    /* Not last employee , allow deletion   */

                    DELETE              EMPLOYEE
                    WHERE               EMPLOYEE : EmployeeNumber = old : EmployeeNumber


          ELSE


                    /* Send a message to the user saying cannot delete last
                   employee in a department .  */

          END IF ;

END ;
```

Class Exercise

- Concerts (PerformerID, ArenaID, ConcertDate, TicketPrice)

- Define a trigger: if inserted price is below 25, print a message and change the ticket price to 25.
- Insert rows to test the trigger

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*
  AS *proc_code*

- exec *proc_name* [@param1 =]value1, …

Stored Procedure Example

- Performers (PerformerID, PerformerName, Street, City, State, Zip)
- Procedure: Insert a performer only if same name and zip not already in the table

Performers (PerformerID, PerformerName, Street, City, State, Zip, ActivityID)

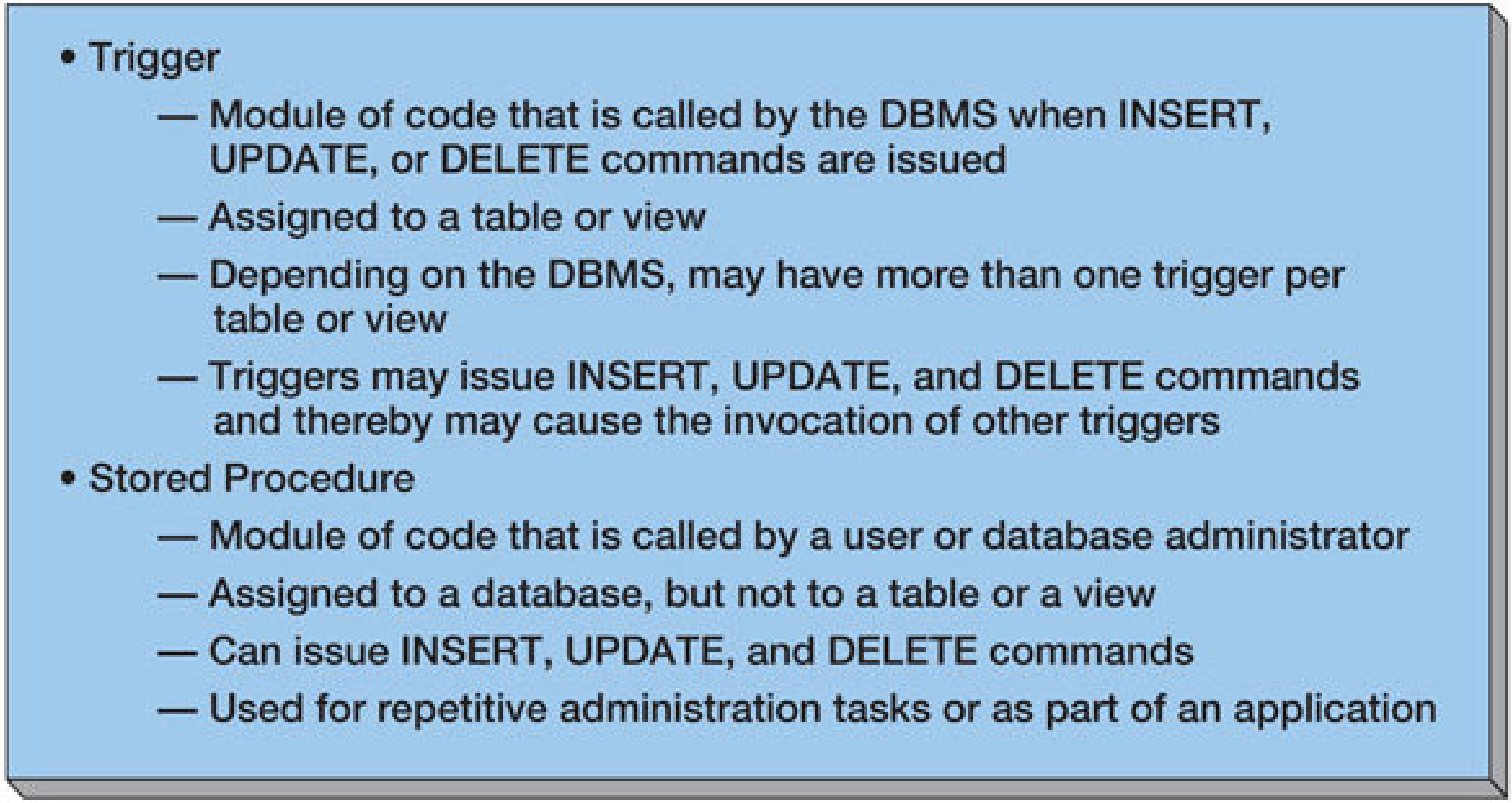
```
CREATE PROCEDURE performer_Insert
    @ID int,
    @NewName  char(20),
    @street char(20),
    @city char(15),
    @state char(2),
    @NewZip  int
AS

    DECLARE @Count as int

    SELECT @Count = Count(*)
    FROM  Performers
    WHERE PerformerName
    =@NewName AND Zip = @NewZip

IF @Count > 0
    BEGIN
        PRINT 'Performer is already in the
    Database'
        RETURN
    END

BEGIN TRANSACTION

    INSERT INTO
    Performers(PerformerID,
    PerformerName, Street, City, State,
    Zip) VALUES (@ID, @NewName,
    @street, @city, @state, @NewZip)

    PRINT 'Performer added to database'

COMMIT
```

To run: exec performer_Insert @ID = 10, @NewName = 'James Brown', @street ='Main', @city ='Aiken', @state ='SC', @NewZip = 54322

Class Exercise

- Add code to the previous procedure to prevent anyone named ‘Spears’ to be inserted into the DB. Print an error explicative message when that happens.
- Test the procedure (exec ….)

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

Class Exercise

- Students(Alpha, LName, FName, GPA)
- Enroll(Alpha, CourseID, Semester, Grade)
- GradeValues(LetterGrade, PointValue)

- Define a trigger to update the GPA every time the student gets a new grade, or a grade changes