Exercise: Stored Procedures

Hint: You can use the manual „MySQL 5.0 Stored Procedures“ by Peter Gulutzan to help you write some stored procedures.

1. Write a stored function that computes some value (for instance to evaluate some mathematical formula like computing the area of a circle, given the radius).

2. Write a Stored Procedure in MySQL, for instance one that computes the yearly salary of a professor in the koch_universitydb, when one provides as input the pName of the professor and under the assumption that the salary in the table refers to monthly values. A function would be better to do this, but MySQL does not support accessing tables from inside functions. Therefore, one needs an OUT parameter to deliver the result.

3. Use a trigger calling a Stored Procedure to sort the students that are newly inserted into two different tables. In particular, do the following steps:

a) Extend the schema of the table student by a column called *stipend* whose values will indicate the type of stipend or scholarship a student may have. The default value will be NULL.

b) Define a new table named *EStudent* to collect students with an Erasmus stipend. The schema of this new table is the same as the schema of the table Student, only without the attribute stipend (because only Erasmus students will be collected here).

c) Define a Stored Procedure *SortOutEStudents* that takes the attributes of a newly created student as input and finds out if this new student is an Erasmus student. If it is an Erasmus student, insert this student into the table EStudent.

d) Define a BEFORE INSERT trigger for the table student, that calls the procedure SortOutEStudents. The purpose is that before a new student is inserted, it will be checked whether this student is an Erasmus student and should therefore not be inserted into the normal Student table but instead into the ErasmusStudents table.

e) Insert some new students with values of a stipend into the Student table. Test some with an Erasmus stipend and some with a different one (may be NULL).

You will find out that the insert into the Student table happens even if it is an Erasmus student. An Erasmus student will additionnally be inserted into the EStudent table.
To prevent this, the trigger would need to stop the insertion into the student table, but still execute the procedure call to perform the insert into the EStudent table.
In former versions of MySQL, it was possible to write a trigger that executed some statements and then ended abruptly because of a runtime error, without undoing its previous steps. This was possible because there was hardly any exception handling.

Nowadays, when the trigger or the Stored Procedure crashes because of an untreated error, the entire block of code (trigger or Stored Procedure) will not be executed at all. So, this construction is no longer possible.

f) Your additional task:
Find a solution to insert a new student and insert them into either the Student table or the EStudent table, depending on whether they have an Erasmus stipend or not.

4. Extend the Java program JDBCTestExample.java from a previous exercise to include an additional menu point that will call a stored procedure that you have defined in the database, for instance to perform the following task:
Ask the user for a matNr and then list all classes for which this student works as a TA, including the hours and salary of each contract.