



# Information Systems: Design and Development

## Information Systems: Design and Development

---

### **Degree in Computer Engineering**

---

### Practice 4

---

### **Stored Procedures and Transactions**

---

Estimated duration: 2 sessions (4 hours)

## OBJECTIVES

---

- Learn how to manage different panels within a *JFrame*
- Learn how to make calls to stored procedures and functions using JDBC
- Learn how to handle transactions

### Panel Management

Create the **Trainer** and **TrainerDAO** classes and create a panel for trainer management

### Stored procedures and functions

- Create a store procedure with 2 parameters:
  - An input parameter for reading an activity identifier
  - An output parameter to return a cursor

This procedure receives an activity identifier and returns, as an output parameter, a cursor with the name and email of those members who are enrolled in that activity.

- Add a new item (Activities) to the main menu with an item (Activity Management)
- Create the **Activity** and **ActivityDAO** classes for activity management (activityDao should currently only have the method that calls the stored procedure)
- Create a new dashboard for Activity Management with the following components:
  1. A *Jtable* to display the result of the procedure
  2. An field to read an activity id (you can replace this field later – if you want - with a *ComboBox* with the name of all the activities in the database)
  3. A button to launch the procedure

### Transaction management

This database schema is designed to not allow the deletion of trainers in charge of some activity. In this case it makes no sense to do a "cascade" deletion since the elimination of a responsible trainer would imply the elimination of the activities for which it is responsible and, therefore, would be lost from the database.

To maintain the "business rule" of the ER model that indicates that an activity must have, compulsorily, a responsible trainer, design the following sequence of operations to be able to eliminate a trainer:

- Check if the trainer oversees any activity. If she/he is not, then delete it directly from the database. If the trainer is in charge of and activity then:
  1. Insert a fictitious trainer with code "M999", named "Generic Trainier" and with DNI "00000000A" (these are the only two mandatory fields in the TRAINER table). This should only be done when this fake trainer does not exist.

2. Change for all those rows in ACTIVITY table in which the trainer oversees an activity, the foreign key for the value "M999"
3. Delete the trainer in trainer table.

Modify the code of the "Delete Trainer" event for a transaction that performs the above operations in a single block.