ETSI
ESCUELA TÉCNICA
SUPERIOR DE INGENIERÍA

Information Systems – Design & Development

Session 4

Stored Procedured Calls

2018/19

2 sessions (4 hours)

Objetives

- `Using JDBC CallableStatements to Execute Stored Procedures`

In many cases, it is more convenient to define the business rules in the DBMS itself. In Oracle, the most complex business rules can be realized through procedures, functions and triggers. This frees the programmer from having to perform operations in the application layers.

Exercise 1

Create a PL / SQL function with the following header:

fExpertS (character) RETURN integer

to do the following:

-The character parameter can receive "M" or "F".
-The function will return the number of male or female experts depending on the character that has been passed by parameter.
-The stored PL / SQL function will throw an exception if the parameter does not correspond to any of the characters indicated above. This exception must be captured in the application layer.

In the ExpertManager class (persistence layer), create a method that calls the stored function:

int sExpert (character)

This function receives as parameter the characters "M" or "F" and returns what the stored function returns.

Finally, in the application layer, design a window (user interface)
in which you can request the characters "M" or "F" and display the result in the form of a label.
Maybe you need to add somethings to the main menu.

Exercise 2

Write a PL / SQL procedure with the following header:

pColaboratesCase (pCodCase, cursorCC)

to do the following:

- The pCodCase parameter corresponds to a case code. It is an input parameter.
- The cursorCC parameter (cursor for Contributors-Case) is output. It must be of type "SYS_REFCURSOR" and it will store the code, the name and the specialty of the experts that have participated in the case with code pCodCase, as well as the description of the collaboration that has been carried out.

In the persistence layer, create a method that calls this procedure and returns a "r*esultSet*" corresponding to the cursor stored in the cursorCC parameter.

```
resultSet listColaboratorsByCase (codCase)
```

Finally, in the application layer, perform the following extensions:

- Add the submenu "Collaboration Management" to the "Collaborations" menu
- Design a new window, which opens when selecting the "Collaboration Management" submenu, in which you can request the code of a case and show the result in the form of a table.