

## Exercicises Procedures IV: Cursors

### EXERCISE 1:

Create a procedure that backs up all the tables in the world DB including the data. The name of the new tables must include the date of the current day in the style of 2020-02-13 using CURDATE().

### EXERCISE 2:

Go to the Northwind DB.

Look at the orders table.

Create a new table called orders\_bck with the same structure as the orders table plus a column called bck\_date of type DATETIME.

Create a procedure that uses a cursor to go through the entire orders table row by row and inserts the files inside the orders\_bck table attaching the current date to the last column.

```
CREATE PROCEDURE backUpOrders() ...
```

### EXERCISE 3:

Aneu a northwind.

Create a procedure that shows the shipper data one by one per screen (opening a result window for each shipper) using a cursor that captures the data from all the columns of the table.

```
CREATE PROCEDURE showShippers() ...
```

### EXERCISE 4:

Go to the northwind DB

Look at the table customers.

Make a procedure with a cursor that checks if a CustomerID passed as a parameter exists. If it does exist, return its ContactName.

If it does not exist, it will show a message informing that it does not exist.

This procedure must have two parameters (1 input and 1 output).

```
CREATE PROCEDURE checkCustomerID(IN vCustomerID VARCHAR(???), OUT  
vContactName VARCHAR(?))
```

### EXERCISE 5:

Use the mymovies database.

Create a new procedure that using cursors shows on screen the title and the year of creation of all the movies that have been created within a period of years specified by the user and ordered by date of creation and title of the movie.

The call for example could be: CALL getMoviesByYear(1986,2001);

With these hints, you know how to say the procedure and what kind of parameters it must have.

#### EXERCISE 6:

Use the mymovies database.

Create a new procedure that using cursors shows on screen the title and the year of creation of all the movies that in their movie title contain the text passed as a parameter.

The call for example could be: `CALL getMoviesByName('Superman');`

With these hints, you know how to say the procedure and what kind of parameters it must have.

#### EXERCISE 7:

Use the mymovies database.

Create a new procedure that using cursors allows the user to calculate the benefits of a specific movie by specifying its movie id.

The profits of the movies are calculated by multiplying the stockUnits \* price column.

The profits of the movie must be stored in an output variable.

This procedure has only one parameter.

Its call for example could be:

```
SET @X = 36;  
CALL calculateRevenue(@X);  
SELECT @X;
```

With these hints, you know how to say the procedure and what kind of parameters it must have.

#### EXERCISE 8:

We will create a procedure in the northwind DB with the objective of generating and exporting a report for each country with the summary of the orders created, the variety of products ordered, the total number of units ordered and the total profit generated.

Creates a procedure called `exportOrderDetailsPerCountry()` without parameters.

Creates a `curCountries` cursor with only the name of the different countries present in the `northwind.customers` table.

For each iteration of the cursor, it generates a file named `'Country_OrderDetailsResum.csv'` where `'Country'` will be changed to the country name in each case. Therefore, we will use dynamic sentence.

These exported files will contain a row with the following columns with the information of:

- a) Country Name AS Country
- b) Request of the requested orders AS QuantitatComandes (hint: `northwind.orders.orderid`)
- c) Recompte de la varietat de diferents productes demanats AS QuantitatProductesDiferents (hint: `northwind.orderdetails.productid`)

- d) Sum of total units ordered among all the orders of each country AS QuantitatUnitats (hint: northwind.orderdetails.quantity)
- e) Sum of total profit generated between all the orders of each country AS ImportTotalComandes (hint: northwind.orderdetails.UnitPrice)

The fitxer will separate the fields by ';', optionally enclose the texts with '""' and separate the files by 'œ'.

#### HINTS:

- 1) First find the different countries in the table northwind.Customers (I already tell you that there are 21).
- 2) To make the cursor you have to use several tables joined with JOINS. Look at which tables you need to determine how many orders have been made from each country and where the details of each transaction are stored.