Facultad de Informática. Computer Science Engineering Bachelor's Degree.
Databases 2018-2019. Group I. Exercises: PL/SQL and triggers.

Exercise 1. You must use for this exercise the DB of *All the Books* bookstore (*script* `05sql2ExBookstore.sql`).

a. Bookstore employees must know the list of books not sent yet.

   Write a **PL/SQL stored procedure** (name it `OrderBooks`) that, given an order Id received as a parameter, shows the list of books requested in that order (ISBN, title, number of copies).

b. Write a stored procedure without parameters named `PendingOrders` that shows on the following information regarding orders not sent yet: client name, address and order Id. After this line of text, it must invoke procedure `OrderBooks` to show the books requested in that order. The list of orders must be sorted chronologically (by order date).

Exercise 2. Create the following tables:

```
CREATE TABLE Contracts(
  Ref VARCHAR(10) PRIMARY KEY,
  Organization VARCHAR(100),
  ContDate DATE,
  NumRoutes NUMBER(2,0));

CREATE TABLE Routes(
  Ref VARCHAR(10) REFERENCES Contracts ON DELETE CASCADE,
  Origin VARCHAR(50),
  Destination VARCHAR(50),
  Vehicle VARCHAR(20),
  PRIMARY KEY (Ref, Origin, Destination));
```

a. Write a stored procedure with a reference contract as input parameter. The procedure must update the information in `NumRoutes`, according to the number of routes associated with the contract. In addition, the procedure must print on the console the name of the organization and the total number of associated routes. You must declare an exception that is thrown to show a message if the reference does not have associated any route.

b. Create a trigger to keep updated the value of `NumRoutes` whenever a row is inserted to or deleted from table `Routes` (suppose that before executing the trigger the value for this column is consistent.)

Exercise 3. Create the following tables:

```
CREATE TABLE Departments (
  CodDept CHAR(5) PRIMARY KEY,
  Name VARCHAR(100));

CREATE TABLE Employees (
  SSN CHAR(9) PRIMARY KEY,
  Name VARCHAR(100),
  CodDept CHAR(5) REFERENCES Departments ON DELETE SET NULL,
  Salary NUMBER(4,0));

CREATE TABLE Changes(
  IdChange VARCHAR(10) PRIMARY KEY,
  UserId VARCHAR(20),
  OldSalary NUMBER(4,0),
  NewSalary NUMBER(4,0));
```

a. Write a trigger that records in the table Changes any update of the salary of the employees. The trigger must store the user, the date of the change, and both the salary before the change and the updated salary. The ID will be obtained from a sequence called `SEQChanges`.

b. Write a stored procedure that lists for each department the name and salary of each employee whose salary is lower than the average of the department. For each department the procedure must show the total amount of these salaries by department.

Exercise 4. Create the following tables:

```
Create table Author (
  SSN CHAR(9) PRIMARY KEY,
  Name VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL;
  Country VARCHAR(30) NOT NULL;
  NumArticles CHAR(3) NOT NULL);

Create table Journal (
  ISSN VARCHAR(9) PRIMARY KEY,
  Name VARCHAR(100) NOT NULL);

Create table Article (
  DOI CHAR(30) PRIMARY KEY,
  Title VARCHAR(100) NOT NULL,
  ISSNJournal VARCHAR(9) NOT NULL REFERENCES Journal(ISSN) ON DELETE CASCADE,
  NumAuthors NUMBER(1,0) NOT NULL,
  CHECK NumAuthors BETWEEN 1 AND 4);

Create table Sign (
  SSN CHAR(9) NOT NULL REFERENCES Author,
  DOI CHAR(30) NOT NULL REFERENCES Article ON DELETE CASCADE,
  PRIMARY KEY(SSN, DOI));
```

a. Write a stored procedure with a journal name as input parameter. The procedure must show the data associated to the journal (`ISSN, Name`) and the names and surnames of the authors that have signed at least one article published in the journal. In the case that the journal has not associated any article, the procedure will show the message: `'No Authors'`.

b. Write a trigger that updates the column `NumAuthors` in the table `Article` whenever an author is inserted, deleted or updated.