



## Examen -- Control 1º -- Ampliación de Bases de datos

----- Marzo, 2020

APELLIDOS:

NOMBRE:

1. (2 puntos) Dada la tabla **Acceso** (**DNIalumno**, **NombreSoftware**, **NúmeroLaboratorio**), que indica los programas a los que tiene acceso cada alumno y los laboratorios en los que está autorizado trabajar dicho alumno. Todos los programas están instalados en todos los laboratorios. Se pregunta:
  - a) ¿Qué dependencias problemáticas hay?
  - b) ¿Porqué son problemáticas?
  - c) ¿Qué anomalía hay cuando se le autoriza al alumno **1234568J** a trabajar con un programa nuevo **Prog1**. Da un ejemplo con filas que muestre la anomalía.
  - d) ¿Cómo la solucionas?
2. (1,5 puntos)
  - a) ¿Qué comprobaciones se usan para decidir si una relación está en Forma Normal de Boyce-Codd (FNBC)?
  - b) Comprueba si la relación  $R(A,B,C,D)$  con las DFs= $\{A \rightarrow C, C D \rightarrow A, C D \rightarrow B\}$  está en FNBC. Explica los pasos.
3. (1,5 puntos) Tenemos la relación, descrita anteriormente, **Acceso** (**DNIalumno**, **NombreSoftware**, **NúmeroLaboratorio**) y otra de **Alumno** (**DNI**, **Apellidos**, **Nombre**, **Dirección**). Cada vez que un alumno accede a un programa en un laboratorio, queremos comprobar si se ha excedido en el número máximo permitido de programas y el número máximo de laboratorios:
  - a) Indica qué técnica de desnormalización nos conviene aplicar para agilizar esa consulta y explica *cómo aplicarla*.
  - b) ¿Qué anomalías de actualización puede provocar esa técnica?
  - c) ¿Cómo solucionas las anomalías?
4. (5 puntos) Usando PLSQL y PLSQL Dinámico
  - a) Escribe un trigger que, cada vez que haces una operación (crear, borrar o modificar cada fila) en la tabla **Fabrica** (**IdFábrica**, **NombreEmpresa**, **totalEmpleados**, **áreaFabricación**), escriba en consola "*se ha hecho la operación LITERAL para NOMBREEMPRESA*". Donde: LITERAL es 'crear fila', 'borrar fila', 'modificar fila', dependiendo de la operación realizada; y NOMBREEMPRESA es el nombre de la empresa de la fila en la que se ha hecho la operación. (no trates excepciones)
  - b) Tenemos una tabla de contabilidad para cada fábrica: **contabilidad\_XXXXX**(**numMovimiento**, **TIPO**, **importe**), donde XXXXX es el IdFabrica, y TIPO es 'deber' si importe es menor de cero y 'haber' en caso contrario. Completa el código (ver debajo) del procedimiento **CrearFilas**: debe insertar una fila en la tabla de contabilidad de la fábrica correspondiente por cada fila en la tabla **movtos**(**numMovimiento**, **IdFabrica**, **importe**). La lectura de esta tabla ya está en el código con el cursor.  
Antes de insertar cada fila, compruebo que exista la tabla de contabilidad para esa fábrica (ya está en el código). Si no existiera la tabla, doy un mensaje indicando que no existe esa tabla y continúo con el siguiente movimiento.

Create o replace procedure **CrearFilas** (**dni VARCHAR**) as

```
CURSOR movtos IS.  -- leo todas las filas
    select numMovimiento, IdFabrica, importe
    from movtos;
si_esta INT;
unmovto movtos%TYPE;

BEGIN

    -- compruebo que la tabla contabilidad_XXXXX existe
    select count(table_name) into si_esta from tabs where
        table_name = RTRIM(UPPER('contabilidad_' || unmovto.IdFabrica));

END;
```