

NOTA: ES MUY IMPORTANTE que siga exactamente la especificación de este ejercicio y que llame a cada objeto **EXACTAMENTE** como se pide.

- Para comenzar el examen conéctese a la dirección: <https://docenciabd.lcc.uma.es/>
- Utiliza el usuario que comienza por UBDXXX para conectarte a la aplicación. La aplicación suministrará un usuario y contraseña con la que debes conectarte a Apolo en Olimpia.
- Tenga cuidado con las comillas si va a copiar y pegar del enunciado, puesto que a veces el código ASCII del pdf no es el mismo que se utiliza en SQL.
- **NO OLVIDE** finalizar las transacciones.
- **AL FINALIZAR EL EXAMEN**, podrá comprobar una nota **PROVISIONAL** de su ejercicio ejecutando las siguientes sentencias:

```
set serveroutput on
begin
CORRIGEBD.corrige_pl_2019;
end;
```

Encontrará en el esquema que se le ha proporcionado una copia de algunas de las tablas con las que ha hecho el trabajo en grupo.

Además, ya se ha creado una tabla denominada `CONTROL_NOMINA` con los atributos

```
Fecha DATE NOT NULL,
Usuario VARCHAR2(35),
Empleado Number NOT NULL,
Antes Number,
Despues Number
```

Ejercicio 1 (0,5 puntos). Crea un Trigger denominado `TR_NOMINA` que cuando se modifique o inserte un registro de/en la tabla `NOMINA`, inserte un registro en `CONTROL_NOMINA` con la fecha del sistema, el usuario que ha realizado la operación (el `USER`), el código del empleado, el valor que tenía antes su `importe_bruto` y el valor nuevo que tiene el `importe_bruto`. Si lo que ha ocurrido es una inserción en `NOMINA`, entonces el atributo `Antes` estará a `NULL`.

Crear el siguiente paquete:

```
CREATE OR REPLACE PACKAGE PK_NOMINAS AS

PROCEDURE P_CREA_NOMINA (FECHA_ACTUAL DATE, MES VARCHAR2);

PROCEDURE P_BORRA_NOMINA (MES VARCHAR2);

PROCEDURE P_VISTAS_EMPLEADOS (Categoria Number);

END;
```

Crea el BODY del paquete con:

Ejercicio 2 (0,5 puntos). Crea en el paquete un procedimiento denominado `P_CREA_NOMINA` (`Fecha_actual date`, `Mes Varchar2`) que inserta en la tabla `NOMINA` una copia de los valores de las nóminas emitidas en el Mes pasado como parámetro pero cambiando la `fecha_emision` por `fecha_actual`. Así, por ejemplo:

`P_CREA_NOMINA ('10/06/2019', '05/2019')` busca las nóminas que se emitieron en mayo de 2019 y las inserta en `NOMINA` pero poniendo como fecha de emisión '10/06/2019'. Recuerda que `TO_CHAR(fecha, 'mm/yyyy')` devuelve el mes y el año de una fecha.

Ejercicio 3 (0,25 puntos). Crea en el paquete un procedimiento denominado `P_BORRA_NOMINA` (`Mes Varchar2`) que borra todas las nóminas del mes que se pasa como parámetro. El mes tendrá el formato 'mm/yyyy'

Ejercicio 4 (0,5 puntos). Crea en el paquete un procedimiento denominado `P_VISTAS_EMPLEADOS` (`Categoria Number`) que crea una vista denominada `USUARIO_NOMINA` para cada uno de los empleados de esa categoría seleccionando sólo sus nóminas. Para ello obtendrá primero el ID del empleado de la tabla `NOMINA` y creará la vista. Así, por ejemplo, si la categoría seleccionada es 4 y hay un empleado de categoría 4 con usuario `PEPITO` que tenga un ID =99 y otro `JUANITO` con ID=100, se crearán dos vistas:

```
CREATE OR REPLACE VIEW PEPITO_NOMINA AS SELECT DNI, NOMBRE, APELLIDO1,
APELLIDO2, FECHA_EMISION, IMPORTE_NETO, IMPORTE_BRUTO FROM EMPLEADO E JOIN
NOMINA N ON E.ID = N.EMPLEADO WHERE E.ID = 99
```

```
CREATE OR REPLACE VIEW JUANITO_NOMINA AS SELECT DNI, NOMBRE, APELLIDO1,
APELLIDO2, FECHA_EMISION, IMPORTE_NETO, IMPORTE_BRUTO FROM EMPLEADO E JOIN
NOMINA N ON E.ID = N.EMPLEADO WHERE E.ID = 100
```

Ejercicio 5 (0,25 puntos). Modifica el procedimiento `P_VISTAS_EMPLEADOS` de modo que cuando no pueda crear la vista correspondiente a un empleado, controle la excepción, muestre un mensaje de error y siga con el resto de empleados de esa categoría. Para probarlo modifica el usuario de un empleado de modo que comience por un número, p.e. `123Pepe`. De esta forma la instrucción `CREATE OR REPLACE VIEW 123Pepe_NOMINA ...` debe dar error puesto que un identificador no puede empezar por número.