

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

- El examen podrá realizarlo en su usuario de práctica habitual (en Afrodita).
- 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.
- Hay que entregar en el CV un archivo plano con formato txt que contenga el código del trigger y el código del CUERPO del paquete PL/SQL.

Ejercicio PLSQL1:

Vamos a implementar un borrado lógico, para ello, dada la tabla:

```
CREATE TABLE tabla_amigo (  
  nombre VARCHAR2(64) NOT NULL,  
  telefono VARCHAR2(32) NOT NULL,  
  borrado CHAR(1) DEFAULT 'N' not null);
```

Y la vista

```
CREATE VIEW amigo AS SELECT nombre , telefono FROM tabla_amigo WHERE borrado = 'N';
```

Crear un trigger de sustitución denominado BORRA_AMIGO sobre la vista AMIGO que al borrar una fila lo que haga sea simplemente poner para dicha fila el atributo BORRADO a 'S' en TABLA_AMIGO;

Ejercicio PLSQL2:

Se desea realizar un sistema capaz de renombrar las restricciones de clave foránea de las tablas de nuestro esquema. Para ello el sistema debe localizar todas las claves foráneas de nuestro esquema y renombrarlas.

El renombrado para las claves foráneas deberá seguir el siguiente formato:

- FK<NOMBRE_TABLA_ORIGEN><NOMBRE_TABLA_DESTINO>

Para ello vamos a realizar:

- Un paquete llamado PAQ_RENOMBRA que contendrá todas las funciones y procedimientos indicados en este ejercicio.
- Un procedimiento denominado RENOMBRA_FK que recibe como primer argumento el nombre de una tabla, como segundo un nombre de una de sus claves foráneas y como tercero el nuevo nombre. El procedimiento se limita a renombrar la clave.
- Una función denominada NUEVO_NOMBRE_FK que recibe como argumento el nombre de una tabla origen, el nombre de una tabla destino y devolverá el nuevo nombre (esta función únicamente genera el nuevo nombre, no debe renombrar nada).

- Un procedimiento denominado RENOMBRA_TODAS_FK que procederá a localizar todas las claves foráneas del usuario y renombrarlas (usando obligatoriamente la función NUEVO_NOMBRE_FK Y RENOMBRA_FK).

Nota: La sentencia para renombrar una restricción sigue el siguiente formato:

```
alter table <tabl_name> rename constraint <old_cons_name> to <new_cons_name>;
```

Ejemplo:

```
CREATE TABLE TDEP
```

```
(CODIGO VARCHAR2(16) PRIMARY KEY, NOMBRE VARCHAR2(16), DIRECTOR VARCHAR2(16));
```

```
CREATE TABLE TEMP
```

```
(NSS VARCHAR2(16) PRIMARY KEY, NOMBRE VARCHAR2(16), DEPARTAMENTO VARCHAR2(16));
```

```
ALTER TABLE TDEP ADD
```

```
(CONSTRAINT FK_REN_1 FOREIGN KEY(DIRECTOR) REFERENCES TEMP(NSS));
```

```
ALTER TABLE TEMP ADD
```

```
(CONSTRAINT FK_REN_2 FOREIGN KEY(DEPARTAMENTO) REFERENCES TDEP(CODIGO));
```

La restricción FK_REN_1 debería quedar renombrada como FKTDPTTEMP

Y la restricción FK_REN_2 debería quedar renombrada como FKTEMPTDEP