

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 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.

En el usuario que se te ha creado ejecuta:

```
CREATE TABLE T_CONTROL (TABLA VARCHAR2(64) NOT NULL, FECHA DATE NOT NULL);
```

```
CREATE TABLE T_AUDITA (USUARIO VARCHAR2(64) NOT NULL, TABLA VARCHAR2(64) NOT NULL, FECHA DATE NOT NULL);
```

Para Probar tu código, además puedes ejecutar:

```
CREATE TABLE MARCA
(IDMARCA NUMBER PRIMARY KEY, NOMBRE VARCHAR2(128), PRECIOHORA
NUMBER(6,2)) ;
CREATE TABLE MODELO
(IDMODELO NUMBER PRIMARY KEY,
IDMARCA NUMBER REFERENCES MARCA (IDMARCA),
NOMBRE VARCHAR2(128),
NUMPUERTAS NUMBER,
COMBUSTIBLE VARCHAR2(64),
CAPACMALETERO NUMBER ) ;
CREATE TABLE CLIENTE
(IDCLIENTE VARCHAR2(16) PRIMARY KEY,
TELEFONO NUMBER,
NOMBRE VARCHAR2(64),
APELLIDO1 VARCHAR2(64),
APELLIDO2 VARCHAR2(64),
EMAIL VARCHAR2(64) ) ;
CREATE TABLE VEHICULO
(NUMBASTIDOR VARCHAR2(64) PRIMARY KEY,
MATRICULA VARCHAR2(16) UNIQUE,
FABRICACION NUMBER,
IDMODELO NUMBER REFERENCES MODELO (IDMODELO),
CLIENTE_IDCLIENTE VARCHAR2(16) REFERENCES CLIENTE,
KILOMETRAJE NUMBER ) ;
CREATE TABLE SERVICIO
(IDSERVICIO NUMBER PRIMARY KEY,
ESTADO VARCHAR2(64) NOT NULL,
FECAPERTURA DATE,
FECRECEPCION DATE,
FECREALIZACION DATE,
OBSCHAPA VARCHAR2(1024),
NUMBASTIDOR VARCHAR2(64) REFERENCES VEHICULO,
IDCLIENTE VARCHAR2(16) REFERENCES CLIENTE ) ;
```

Crear el siguiente paquete:

```
CREATE OR REPLACE PACKAGE PAQ_CREA AS
ERROR_TABLA EXCEPTION;
FUNCTION NUM_COLUMNAS (P_TABLA VARCHAR2) RETURN NUMBER;
FUNCTION NUM_REFERENCIAS (P_TABLA VARCHAR2) RETURN NUMBER;
PROCEDURE CREA_VISTAS_COLUMNAS (MIN_COLUMNAS NUMBER);
FUNCTION NUM_TABLAS_REFERENCIADAS (P_TABLA VARCHAR2) RETURN NUMBER;
END;
```

Ejercicio 1 (0,25 puntos). Crear un trigger denominado **TR_CONTROL** para que al insertar una fila en T_CONTROL almacene en T_AUDITA el nombre del usuario actual (USER), así como los datos insertados (TABLA y FECHA).

Crea el BODY del paquete con:

Ejercicio 2 (0,25 puntos). Crea en el paquete la función **NUM_COLUMNAS** que recibe un único argumento, el nombre de una tabla, y devuelve el número de columnas de la tabla. Si la tabla no existe, debe elevar la excepción **ERROR_TABLA**.

Ejercicio 3 (0,5 puntos). Crea en el paquete la función **NUM_REFERENCIAS** que recibe un único argumento, el nombre de una tabla, y devuelve el número de FOREIGN KEYS de la tabla. Si la tabla no existe, debe elevar la excepción **ERROR_TABLA**. Así, por ejemplo, NUM_REFERENCIAS ('VEHICULO') devuelve 2. Pista: utiliza user_constraints

Ejercicio 4 (0,5 puntos). Crea en el paquete el procedimiento **CREA_VISTAS_COLUMNAS** que se recorre las tablas del usuario que tengan, al menos, tantas columnas como MIN_COLUMNAS. Para cada tabla encontrada:

- Inserta en T_CONTROL el nombre de la tabla y la fecha del sistema
- Crea una vista con el nombre de la tabla y el prefijo VC_. Por ejemplo, si la tabla se llama TABLA1, ejecuta `CREATE VIEW VC_TABLA1 AS SELECT * FROM TABLA1.`

Ejercicio 5 (0,5 puntos). Crea en el paquete la función **NUM_TABLAS_REFERENCIADAS** que recibe un único argumento, el nombre de una tabla, y devuelve el número de tablas distintas y distinta a ella misma que la referencian mediante claves foráneas. Nótese que no se pide contar las restricciones de clave foránea, sino las tablas distintas implicadas. Esta función devolverá la excepción **ERROR_TABLA**, en caso de que la tabla no exista.

➤ Pista. Puede que te sirva la sentencia:

```
select  a1.table_name,a2.table_name  from  user_constraints  a1,
user_constraints      a2      where      a1.r_constraint_name      =
a2.constraint_name;
```