

# Práctica final PL/SQL 12c

## 1. Objetivos

- Esta práctica pretende hacer un repaso de los componentes más importantes de PL/SQL: procedimientos, funciones, paquetes y triggers.

## 2. Crear las tablas y datos necesarios

- Vamos a crear cuatro tablas. Podemos ejecutar el script “**creación\_ddl.sql**” que se encuentra en los recursos del capítulo o bien podéis copiarlo y pegarlo de este documento

```

FACTURAS
LINEAS_FACTURA
LOG_CONTROL
PRODUCTOS

```

- Las columnas de cada tabla con la siguientes:

### FACTURAS

|                    |                      |
|--------------------|----------------------|
| <b>COD_FACTURA</b> | <b>NUMBER</b>        |
| <b>FECHA</b>       | <b>DATE</b>          |
| <b>DESCRIPCIÓN</b> | <b>VARCHAR2(100)</b> |

- Clave primaria: COD\_FACTURA

### LINEAS\_FACTURAS

|                     |               |
|---------------------|---------------|
| <b>COD_PRODUCTO</b> | <b>NUMBER</b> |
| <b>COD_FACTURA</b>  | <b>NUMBER</b> |
| <b>PVP</b>          | <b>NUMBER</b> |
| <b>UNIDADES</b>     | <b>NUMBER</b> |
| <b>FECHA</b>        | <b>DATE</b>   |

- Clave Primaria: COD\_FACTURA+COD\_PRODUCTO
- Referencia a la tabla FACTURAS y a la tabla PRODUCTOS

## PRODUCTOS

|                        |                      |
|------------------------|----------------------|
| <b>COD_PRODUCTO</b>    | <b>NUMBER</b>        |
| <b>NOMBRE_PRODUCTO</b> | <b>VARCHAR2(100)</b> |
| <b>PVP</b>             | <b>NUMBER</b>        |
| <b>TOTAL_VENDIDO</b>   | <b>NUMBER</b>        |

- Clave primaria: COD\_PRODUCTO

## CONTROL\_LOG

|                       |                     |
|-----------------------|---------------------|
| <b>COD_EMPLEADO</b>   | <b>NUMBER</b>       |
| <b>FECHA</b>          | <b>DATE</b>         |
| <b>TABLA_AFECTADA</b> | <b>VARCHAR2(50)</b> |
| <b>COD_OPERACIÓN</b>  | <b>CHAR(1)</b>      |

- La columna COD\_OPERACION debe valer:(I, U, D → INSERT, UPDATE ,DELETE)

## SCRIPTS DE CREACIÓN

- Los scripts son los siguientes:

```

-----
-- DDL for Table LINEAS_FACTURA
-----

CREATE TABLE "HR"."LINEAS_FACTURA"
( "COD_FACTURA" NUMBER,
  "COD_PRODUCTO" NUMBER,
  "PVP" NUMBER,
  "UNIDADES" NUMBER,
  "FECHA" DATE
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

```

-----  
 -- DDL for Table FACTURAS  
 -----

```
CREATE TABLE "HR"."FACTURAS"
( "COD_FACTURA" NUMBER(5,0),
  "FECHA" DATE,
  "DESCRIPCION" VARCHAR2(100 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

-----  
 -- DDL for Table PRODUCTOS  
 -----

```
CREATE TABLE "HR"."PRODUCTOS"
( "COD_PRODUCTO" NUMBER,
  "NOMBRE_PRODUCTO" VARCHAR2(50 BYTE),
  "PVP" NUMBER,
  "TOTAL_VENDIDOS" NUMBER
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

-----  
 -- DDL for Table CONTROL\_LOG  
 -----

```
CREATE TABLE "HR"."CONTROL_LOG"
( "COD_EMPLEADO" NUMBER,
```

```

"FECHA" DATE,
"TABLA" VARCHAR2(20 BYTE),
"COD_OPERACION" CHAR(1 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
REM INSERTING into HR.LINEAS_FACTURA
SET DEFINE OFF;

-----

-- DDL for Index LINEAS_FACTURA_PK

-----

CREATE UNIQUE INDEX "HR"."LINEAS_FACTURA_PK" ON
"HR"."LINEAS_FACTURA" ("COD_FACTURA", "COD_PRODUCTO")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

----

-----

-- Constraints for Table LINEAS_FACTURA

-----

ALTER TABLE "HR"."LINEAS_FACTURA" ADD CONSTRAINT
"LINEAS_FACTURA_PK" PRIMARY KEY ("COD_FACTURA",
"COD_PRODUCTO")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ENABLE;

```

```

ALTER TABLE "HR"."LINEAS_FACTURA" MODIFY ("COD_PRODUCTO"
NOT NULL ENABLE);

ALTER TABLE "HR"."LINEAS_FACTURA" MODIFY ("COD_FACTURA"
NOT NULL ENABLE);

REM INSERTING into HR.PRODUCTOS
SET DEFINE OFF;
Insert                into                HR.PRODUCTOS
(COD_PRODUCTO,NOMBRE_PRODUCTO,PVP,TOTAL_VENDIDOS) values
('1','TORNILLO','1',null);

Insert                into                HR.PRODUCTOS
(COD_PRODUCTO,NOMBRE_PRODUCTO,PVP,TOTAL_VENDIDOS) values
('2','TUERCA','5',null);

Insert                into                HR.PRODUCTOS
(COD_PRODUCTO,NOMBRE_PRODUCTO,PVP,TOTAL_VENDIDOS) values
('3','ARANDELA','4',null);

Insert                into                HR.PRODUCTOS
(COD_PRODUCTO,NOMBRE_PRODUCTO,PVP,TOTAL_VENDIDOS) values
('4','MARTILLO','40',null);

Insert                into                HR.PRODUCTOS
(COD_PRODUCTO,NOMBRE_PRODUCTO,PVP,TOTAL_VENDIDOS) values
('5','CLAVO','1',null);

```

### 3. Componentes de la práctica

- La práctica pretende realizar los componentes necesarios para gestionar esas tablas. En concreto:
  - Paquete para gestionar las facturas
  - Paquete para gestionar las líneas de factura
  - Triggers para controlar el acceso a las tablas
  -

#### 3.1. PAQUETE FACTURAS

##### PROCEDIMIENTOS

- ALTA\_FACTURA (COD\_FACTURA, FECHA, DESCRIPCIÓN).
  - Debe dar de alta una factura con los valores indicados en los parámetros
  - Debe comprobar que no se duplica
- BAJA\_FACTURA (cod\_factura).
  - Debe borrar la factura indicada en el parámetros
  - Debe borrar también las líneas de facturas asociadas en la tabla LINEAS\_FACTURA.
- MOD\_DESCRI(COD\_FACTURA, DESCRIPCION).
  - Debe modificar la descripción de la factura que tenga el código del parámetro con la nueva descripción
- MOD\_FECHA (COD\_FACTURA, FECHA).
  - Debe modificar la descripción de la factura que tenga el código del parámetro con la nueva fecha
  -

##### FUNCIONES

- NUM\_FACTURAS(FECHA\_INICIO, FECHA\_FIN).
  - Devuelve el número de facturas que hay entre esas fechas
- TOTAL\_FACTURA(COD\_FACTURA.)
  - Devuelve el total de la factura con ese código. Debe hacer el sumatorio de “pvp\*unidades” de las líneas de esa factura en la tabla LINEAS\_FACTURA

#### 3.2. PAQUETE LINEA\_FACTURAS

##### PROCEDIMIENTOS

- ALTA\_LINEA (COD\_FACTURA, COD\_PRODUCTO, UNIDADES, FECHA)
  - Procedimiento para insertar una línea de Factura
  - Debe comprobar que existe ya la factura antes de insertar el registro.
  - También debemos comprobar que existe el producto en la tabla de PRODUCTOS.
  - El PVP debemos seleccionarlo de la tabla PRODUCTOS
- BAJA\_LINEA (cod\_factura, COD\_PRODUCTO)
  - Damos de baja la línea con esa clave primaria)
- MOD\_PRODUCTO(COD\_FACTURA,COD\_PRODUCTO,PARAMETRO)
  - Se trata de 2 métodos sobrecargados, es decir el segundo parámetro debe admitir los siguientes valores:
    - MOD\_PRODUCTO(COD\_FACTURA,COD\_PRODUCTO, UNIDADES)
    - MOD\_PRODUCTO(COD\_FACTURA,COD\_PRODUCTO, FECHA)
  - Por tanto, debe modificar o bien unidades si se le pasa un NUMBER o bien la fecha si se le pasa un DATE

## FUNCIONES

- NUM\_LINEAS(COD\_FACTURA)
  - Devuelve el número de líneas de la factura

### 3.3. TRIGGERS

#### Triggers de tipo sentencia

- Creamos 2 triggers de tipo SENTENCIA, uno para la tabla FACTURAS y otro para la tabla LINEAS\_FACTURA
- Cada cambio en alguna de las tablas (Insert, update, delete), debe generar una entrada en la tabla CONTROL\_LOG con los datos siguientes:
  - Tabla (FACTURAS O LONEAS\_FACTURA)
  - Fecha → usamos la función SYSDATE
  - Usuario que lo ha realizado → función USER
  - Operación realizada (I-U-D)

#### Trigger de tipo fila

- La columna TOTAL\_VENDIDO, de la tabla PRODUCTOS mantiene el total de ventas de un determinado producto.
- Para controlarlo, creamos un Trigger de tipo fila sobre la tabla LINEAS\_FACTURA, de forma que cada vez que se añada, cambie o borre

una línea se actualice en la tabla PRODUCTOS la columna TOTAL\_VENDIDO.

- Si se inserta una nueva línea con ese producto, se debe añadir el total al campo.
- Si se borra la línea debemos restar el total
- Si se modifica, debemos comprobar si el valor antiguo era superior al nuevo y sumamos o restamos dependiendo del resultado