

## EJERCICIOS DE GESTIÓN

1. Vamos a crear una nueva tabla en la que tengamos el nombre del cliente y todo lo gastado por cada cliente, por lo que vamos a crear la tabla y rellenarlas con los datos adecuados.

```
CREATE TABLE GASTO_CLIENTE AS(  
SELECT c.NOMBRE, SUM(lf.CANT*lf.PRECIO) AS TOTAL_GASTADO  
FROM CLIENTES c, FACTURAS f, LINEAS_FAC lf  
WHERE c.CODCLI = f.CODCLI AND f.CODFAC = lf.CODFAC  
GROUP BY c.NOMBRE);
```

2. Vamos a borrar la tabla anterior, y rellenar los datos, pero sólo para aquellos clientes que hayan tenido tres o más facturas.

```
CREATE TABLE GASTO_CLIENTE AS(  
SELECT c.NOMBRE, SUM(lf.CANT*lf.PRECIO) AS TOTAL_GASTADO  
FROM CLIENTES c, FACTURAS f, LINEAS_FAC lf  
WHERE c.CODCLI = f.CODCLI AND f.CODFAC = lf.CODFAC  
HAVING COUNT(F.CODFAC)>3  
GROUP BY c.NOMBRE);
```

3. Vamos a añadir en provincias un nuevo campo en el que guardaremos el número de pueblos que tiene cada provincia. Rellena el nuevo campo de la tabla.

```
ALTER TABLE PROVINCIAS ADD NUMPUEBLOS NUMBER(3);  
  
UPDATE PROVINCIAS P1 SET NUMPUEBLOS = (SELECT COUNT(p.CODPRO) AS NUM_PUEBLOS  
FROM PUEBLOS p, PROVINCIAS P2 WHERE P.CODPRO = P2.CODPRO AND P1.CODPRO = P2.CODPRO GROUP BY P2.CODPRO);
```

4. Borra los artículos que no se han vendido nunca.

```
DELETE FROM ARTICULOS WHERE CODART NOT IN (SELECT DISTINCT (CODART) FROM LINEAS_FAC);
```

5. Añade dos campos a facturas, en el que se guardará el total de la factura, y el total con iva (recuerda que antes de calcular el iva hay que calcularle el descuento) de cada factura. Estos campos no podrán tener valores nulos. Rellena los campos con los valores que tenemos en la base de datos.

```
ALTER TABLE FACTURAS ADD BASE NUMBER(6,0);
ALTER TABLE FACTURAS ADD TOTAL NUMBER(6,0);

UPDATE FACTURAS f2 SET f2.BASE = (SELECT SUM(lf.CANT*lf.PRECIO) AS BASE
FROM LINEAS_FAC lf, FACTURAS f WHERE f.CODFAC = lf.CODFAC AND f.CODFAC =f2.CODFAC GROUP BY lf.codfac) ;

UPDATE FACTURAS f1 SET f1.total = (SELECT (f2.BASE - NVL(f2.DT0,0))*((NVL(f2.IVA,0)/100)+1) AS total
FROM FACTURAS f2 WHERE f2.CODFAC = f1.CODFAC );
```

6. Añade un nuevo campo en la tabla clientes, que se llame código que no acepte valores nulos, y rellena el campo de forma que almacene la primera letra del nombre, la letra de en medio del nombre, y la última letra del nombre. Si el nombre tiene un número par de letras se tomará la siguiente, y si es impar justo la de en medio.

```
ALTER TABLE CLIENTES ADD CODIGO VARCHAR2(2) NOT NULL;
```

```
UPDATE CLIENTES C SET CODIGO =
(SELECT CONCAT(SUBSTR(NOMBRE, 0, 1), SUBSTR(NOMBRE, (LENGTH(NOMBRE)/2)+1,1)) FROM CLIENTES C2 WHERE
C.CODCLI = C2.CODCLI)
```

7. Vamos a añadir en pueblos un nuevo campo en el que guardaremos el número de clientes que tiene cada pueblo. Rellena el nuevo campo de la tabla con los valores apropiados.

```
ALTER TABLE PUEBLOS ADD NUM_CLIENTES NUMBER(3);

UPDATE PUEBLOS P1
SET P1.NUM_CLIENTES =
(SELECT COUNT(C.CODCLI) FROM PUEBLOS p , CLIENTES c
WHERE p.CODPUE = c.CODPUE AND p.CODPUE =p1.CODPUE
GROUP BY p.CODPUE);
```

8. Borra las provincias que no tengan pueblos, pero sin usar el nuevo campo que hemos añadido anteriormente.

```
*/  
DELETE FROM PROVINCIAS p3 WHERE p3.CODPRO =  
(SELECT P2.CODPRO FROM PUEBLOS p, PROVINCIAS p2  
WHERE P2.CODPRO = P.CODPRO AND p3.CODPRO = p2.CODPRO  
HAVING COUNT(P.CODPUE)=0  
GROUP BY P2.CODPRO )
```

9. Vamos a modificar los nombres de los clientes para que salgan bien, es decir nombre y apellido, en vez de “apellido,nombre”.

```
UPDATE CLIENTES C2 SET C2.NOMBRE =  
(SELECT CONCAT(CONCAT(SUBSTR(NOMBRE, INSTR(C.NOMBRE , ' ', 1, 1)+1) , ' '), SUBSTR(NOMBRE, 1, INSTR(C.NOMB
```

Falta parte del final de la línea superior.

```
FROM CLIENTES c WHERE C2.CODCLI=C.CODCLI )
```

10. Crea una nueva tabla en la que almacenaremos la descripción del artículo y el precio de venta. Rellena esos datos para aquellos artículos que **sólo** se han vendido con un precio igual al de venta.

```
Rellena esos datos para aquellos artículos que sólo se han vendido con u  
CREATE TABLE ARTICULOS_IMPORTE AS  
(SELECT DISTINCT A.DESCRIP, A.PRECIO FROM ARTICULOS a, LINEAS_FAC lf  
WHERE A.CODART = LF.CODART AND A.PRECIO = LF.PRECIO)
```

11. Añade un nuevo campo en facturas, en el que se almacenará el número total de artículos que se han vendido en la factura.

```
ALTER TABLE FACTURAS ADD TOTAL_ARTICULOS NUMBER(3);  
  
UPDATE FACTURAS F SET F.TOTAL_ARTICULOS =  
(SELECT COUNT(lf.CANT) FROM LINEAS_FAC lf, FACTURAS f2  
WHERE F2.CODFAC = F.CODFAC AND LF.CODFAC = F2.CODFAC  
GROUP BY LF.CODFAC)
```

12. Añade un nuevo campo en lineas\_fac en donde se almacenará el valor total de esa línea

```
ALTER TABLE LINEAS_FAC ADD TOTAL_LINEA NUMBER(6);

UPDATE LINEAS_FAC LF SET LF.TOTAL_LINEA =
(SELECT SUM(lf1.CANT*LF1.PRECIO) FROM LINEAS_FAC lf1
WHERE LF.CODFAC = LF1.CODFAC
GROUP BY LF.CODFAC);
```

13. Añade un nuevo campo en la tabla artículo en el que se deberá poner un 0 si el artículo no se ha vendido nunca, un 1 si el artículo alguna vez se ha vendido con un precio inferior al de venta actual, y un dos si el artículo se ha vendido con un precio igual al de venta actual.

```
ALTER TABLE ARTICULOS ADD INDICE_VENTA NUMBER(1);

UPDATE ARTICULOS a SET INDICE_VENTA =
(SELECT
CASE
WHEN (SELECT COUNT(lf.CODART) FROM LINEAS_FAC lf, ARTICULOS a2 WHERE lf.CODART = a2.CODART AND a2.CODART= a.CODART AND a2.PRECIO < a.PRECIO) <> 0 THEN 1
WHEN (SELECT COUNT(lf.CODART) FROM LINEAS_FAC lf, ARTICULOS a2 WHERE lf.CODART = a2.CODART AND a2.CODART= a.CODART AND a2.PRECIO = a.PRECIO) <> 0 THEN 2
END
FROM LINEAS_FAC lf2, ARTICULOS a3
WHERE lf2.CODART=a3.CODART AND a.CODART=a3.CODART
GROUP BY a.CODART);
```

continuación de los “when”

```
CODART AND a2.CODART= a.CODART GROUP BY lf.CODART)=0 THEN 0
ODART AND a2.CODART= a.CODART AND lf.PRECIO < a2.PRECIO GROUP BY lf.CODART)<>0 THEN 1
ODART AND a2.CODART= a.CODART AND lf.PRECIO = a2.PRECIO GROUP BY lf.CODART)<>0 THEN 2
```

14. Crea una nueva tabla en la que almacenaremos la descripción del artículo y el precio de venta. Rellena esos datos para aquellos artículos que se han vendido con un precio igual al de venta, al menos tres veces.

```
CREATE TABLE ARTICULOS_IMPORTE_2 AS
(SELECT DISTINCT a.DESCRIP, a.PRECIO FROM ARTICULOS a, LINEAS_FAC lf
WHERE a.CODART = lf.CODART AND a.PRECIO = lf.PRECIO
AND lf.CODART IN
(SELECT lf.CODART FROM LINEAS_FAC lf HAVING COUNT(lf.CODART)>3 GROUP BY lf.CODART));
```

**Rellena todos los campos y tablas que has tenido que crear con los valores adecuados, y realiza los ejercicios con el número mínimo posible de sentencias sql.**

INSERTS, UPDATES Y DELETES AVANZADOS