

# ESTRUCTURA DE DATOS 2016/2017

## Práctica 2

Andrés Salas Peña  
Rodrigo Alonso De Pool Alcántara

Grupo 1202

### 1. Introducción

Esta práctica consiste en refinar nuestra base de datos de la práctica 1 para que podamos trabajar con ella y manipularla a través de C. Lo primero que se pide es poblar las tablas de nuestra base de datos. Después, modificar las consultas realizadas en la práctica 1 para que funcionen con las modificaciones que hayamos hecho en los apartados anteriores de esta segunda práctica. Y, por último, crear cuatro programas en C que nos permitan acceder a nuestra base de datos y modificarla según lo que pida cada una de las funciones.

### 2. Tablas Diseño

AUTOR		
Nombre	Apellidos	<u>Id_Autor</u>

FIDELIZADO				
Nombre	Screen_name	Tarjeta de Crédito	<u>DNI</u>	Y-N

VENTAS				
Fecha	Efectivo/Tarjeta	Precio Total	<u>Id_Venta</u>	DNI

OFERTAS				
Inicio	Fin	Descuento	<u>Id_Oferta</u>	ISBN

EDICION								
Titulo	Tipo	Cantidad	Titulo_VO	Editor	Costos	Idioma	Precio	<u>ISBN</u>

VENDIDO	
ISBN	Id_Venta

ESCRITO POR	
ISBN	Id_Autor

Los cambios realizados en las tablas fueron los siguientes:

En la tabla FIDELIZADO hemos introducido el screen\_name para que concordara con lo pedido en los programas. También añadimos la columna Y-N (yes / no) que nos permite ver si un usuario ha sido o no borrado de la base de datos sin necesidad de eliminarlo de la misma, ya que eliminarlo supondría una pérdida de información en cuanto a las ventas.

En la tabla EDICION hemos añadido la columna precio que nos permite saber el precio de un libro previo a su compra y aplicar las ofertas directamente sobre este ( consulta que no nos permitía nuestra anterior implementación).

### 3. Comandos Poblar Tablas:

En la implementación de nuestro diseño y tras comentarlo con nuestro profesor, decidimos usar nuestras propias tablas. Nuestro principal motivo fue el hecho de que con nuestras propias tablas podíamos monitorizar de formas más precisa los resultados de nuestros programas y los resultados esperados. Además, evitaba la posible incongruencia de los datos suministrados.

Para poblar nuestras tablas lo primero que hicimos fue realizar los cambios necesarios en nuestra base de datos para que fuera acorde a lo requerido en la práctica y luego creamos los ficheros de texto que contenían información sobre la librería. Después, usando para ello el comando import de pgAdmin (a través del interfaz gráfico facilitado por el programa), insertamos estos ficheros en las correspondientes tablas. Y, por último, a medida que realizábamos consultas y considerábamos posibles casos distintos en sus resultados, usábamos la consulta en SQL de insert into para añadir nuevos registros a nuestras tablas y probar con casos distintos nuestras consultas.

*Nota:*

*Los ficheros de texto están adjuntos en la carpeta DatosNuestros.*

*El backup que hemos adjuntado ya tiene los datos cargados.*

### 4. Consultas SQL y Resultados:

1. Dado un título, ¿Cuántas ediciones tiene? ¿En cuántos idiomas?

```
SELECT COUNT("ISBN")
FROM "Edicion"
WHERE "Titulo_vo"='Game of thrones'
```

	count bigint
1	2

```
SELECT COUNT(P.idiom)
FROM (SELECT DISTINCT "Idioma" as idioma
      FROM "Edicion"
      WHERE "Titulo_vo"='Tokio blues') as P
```

	count bigint
1	1

2. ¿Cuántos libros se han vendido de un autor dado?

```
SELECT    COUNT(V."id_venta")
FROM      "Escrito_por" as S, "Vendido" as V
WHERE     S."Autor_id"='1' AND V."ISBN"=S."ISBN"
```

	count bigint
1	3

3. ¿Cuántos libros de un autor dado se han vendido en oferta?

```
SELECT    COUNT(P."idv")
FROM      (SELECT    VD."id_venta" as idv, V."Fecha" as date
FROM      "Escrito_por" as EP, "Vendido" as VD, "Ventas" as V
WHERE     EP."Autor_id"='1' AND EP."ISBN"=VD."ISBN" AND
          VD."id_venta"=V."id_venta") as P,
          "Vendido" as V, "Ofertas" as O
WHERE     P."idv"=V."id_venta" AND V."ISBN"=O."ISBN" AND
          P."date"<=O."Fin" AND O."Inicio"<=P."date"
```

	count bigint
1	2

4. ¿Cuánto dinero se ha ganado vendiendo libros de un editor dado?

```
SELECT    SUM(L."pp")-SUM(L."cc")
FROM      (SELECT    ED."Precio" as pp, ED."Costos" as cc
FROM      "Edicion" as ED, "Vendido" as VD
WHERE     ED."Editor"='Anaya' AND ED."ISBN"=VD."ISBN")
AS L
```

	?column? double precision
1	99

5. ¿Cuántos libros han comprado los usuarios fidelizados?

```
SELECT    COUNT(*)
FROM      "Ventas" as V, "Vendido" as VD
WHERE     V."DNI" IS NOT NULL AND VD."id_venta"=V."id_venta"
```

	count bigint
1	3

6. ¿Cuántos usuarios fidelizados han comprado libros en inglés?

```
SELECT      COUNT(P."d")
FROM        (SELECT  DISTINCT "DNI" as d
              FROM    "Ventas" as V, "Vendido" as N, "Edicion" as E
              WHERE   V."DNI" IS NOT NULL AND E."Idioma"='Ingles'
                  AND V."id_venta"=N."id_venta"
                  AND N."ISBN"=E."ISBN") as P
```

	count bigint
1	0

Ingles

Español

	count bigint
1	3

EL resultado en nuestra base de datos en ingles es 0 pero si probamos con otro idioma como el español vemos que da el resultado correcto (3)

7. ¿Cuánto dinero se ha ganado vendiendo libros en Francés?

Como en nuestra base de datos nos tenemos libros en francés probamos la misma pregunta pero con libros en Español. El funcionamiento es equivalente para el francés.

```
SELECT      SUM(Ed."Precio")-SUM(Ed."Costos")
FROM        "Edicion" as Ed, "Vendido" as Vd
WHERE       Ed."ISBN"=Vd."ISBN" AND Ed."Idioma"='Espanol'
GROUP BY   Ed."Idioma"
```

	?column? double precision
1	213

8. ¿En que días hubo ofertas de libros de la editorial Adelphi?

Como en nuestra base de datos no tenemos incluida la editorial Adelphi probamos la consulta con la editorial SM.

```
SELECT      O."Inicio", O."Fin"
FROM        "Edicion" as E, "Ofertas" as O
WHERE       E."Editor"='SM' AND O."ISBN"=E."ISBN"
```

	Inicio date	Fin date
1	2011-	2012
2	2014-	2015

### 9. ¿Qué usuarios fidelizados no han comprado nunca libros de bolsillo?

Como en nuestra base de datos no tenemos libros de bolsillo (aunque se podrían agregar en el atributo tipo de Edicion) probamos la misma consultas con tipo 'Blanda'.

```
(SELECT      F."Dni"
FROM        "Fidelizado" as F)
EXCEPT
(SELECT      V."DNI"
FROM        "Ventas" as V, "Vendido" as VD, "Edicion" as E
WHERE       V."id_venta" =VD."id_venta"
AND VD."ISBN"=E."ISBN" AND E."Tipo"='Blanda')
```

	Dni bigint
1	4
2	5
3	3

## 5. Código C

Todo el código C se encuentra adjunto en esta misma entrega. Las pruebas realizadas al código fueron todas satisfactorias, realizando las funciones que se requerían de ellas en la práctica. Además, el código tiene múltiples controles de errores que evitan que un paso de argumentos erróneo tenga un comportamiento inesperado.

Por otro lado al probar con valgrind nos dan pérdidas de memoria que al analizarlas, hemos comprobado que son problemas de la librería odbc y no de nuestra gestión de memoria. Para llegar a esta conclusión creamos un programa que únicamente se ocupaba de crear memoria para un 'statement' de sql y luego lo liberaba, incluso en este caso al pasar valgrind el programa daba múltiples errores de memoria. Esta misma prueba fue realizada con los tests que también nos dieron problemas de memoria. Esto hizo aún más complicado buscar posibles fugas nuestras, si las hubiera.

Nota:

El código está adjunto en la carpeta Codigo.

## 6. Conclusiones Finales

Esta práctica nos ha hecho comprender la cantidad de rediseños que sufren las bases de datos hasta que se logra alcanzar un diseño óptimo. Por otro lado, nos ha permitido entender la dificultad que puede suponer moldear los datos que se tienen a una base de datos dada.

En segundo lugar, nos ha permitido comprender como se opera realmente con bases de datos a través de programas , además de comprender como implementar estos programas para comunicarnos con bases de datos.

Por último, hemos entendido el enorme trabajo que tiene detrás cada una de las bases de datos modernas y también la utilidad de poder trabajar con ellas a través de programas externos (en nuestro caso C) que permiten extender la funcionalidad de las mismas.