

Memoria de la práctica 2

Alejandro Cabana y Víctor García

Introducción

El objetivo principal de esta práctica, además de comprender a un nivel más profundo el funcionamiento de las bases de datos, es ser capaces de conectarse a una base de datos (creada y poblada por nosotros) a través de programas en C, desarrollados por nosotros también, utilizando la librería ODBC.

Poblar las tablas

En esta parte debíamos, a partir de unos ficheros con datos ya proporcionados, poblar las tablas de nuestra base de datos. Para ello, primero debíamos crear unas tablas temporales, una por cada fichero, e insertar los datos de los ficheros en ellas. A la hora de insertar los datos, tuvimos algunos problemas, teniendo que modificar algunos aspectos de los ficheros de datos dados. Por ejemplo, tuvimos que cambiar todas las comas de los precios en isbn_precios.txt por puntos, pues a la hora de insertar los datos en nuestra tabla temporal, al ser la columna precio de tipo double, no aceptaba las comas. Al principio, la columna precio era de tipo character varying, y posteriormente creamos otra tabla temporal llamada precios2 donde cambiamos esa columna al tipo double, escribiendo además en aquellas tuplas que no tenían un valor en precio un 0 en vez de un espacio en blanco.

Además, en LIBROS_FINAL.txt, tuvimos que cambiar algunas fechas que eran 0000-00-00 por 1234-12-12 (arbitrario), pues éstas daban problemas al insertarlas.

A la hora de pasar los datos de las tablas temporales a nuestras tablas de nuestra base de datos, tuvimos que corregir algunas diferencias notables entre ambas. El caso más destacable es el de la columna de número de páginas de nuestra tabla edición, que teníamos que poblar a partir de la columna de número de páginas de la tabla temporal LIBROS_FINAL.txt. Para ello, procesamos por separado las tuplas en las que el número de páginas era un valor erróneo (en algunas ponía libro de bolsillo) y las que tenían un valor numérico seguido de lapalabra pages. Para las primeras, sustituimos por un 0, y para las segundas seleccionábamos la parte numérica y le hacíamos un casting a integer para poder introducir los datos en nuestra tabla edición.

Tablas definitivas de nuestra base de datos (columnas en negrita = claves primarias):

aplica (of_id, isbn)

*autor (**nombre**)*

*cliente (**cl_id**, scrname, full_name, expiration, tarjeta, join_date, borrado)*

*edicion (idioma, editor, precio, **isbn**, formato, numero_de_paginas, fecha_de_publicacion, titulo)*

escribe (autor, libro)

*libro (**titulo**)*

*oferta (descuento, **of_id**, fecha_inicio, fecha_final)*

vende (venta_id, isbn, importe)

*venta (**venta_id**, fecha, cl_id)*

Consultas de la práctica 1

A la hora de probar las consultas realizadas en la práctica 1 en nuestra “nueva” base de datos modificada, tuvimos que realizar varios ajustes. Cambiamos todas las columnas para que no contuvieran mayúsculas, por lo que no era necesario escribir el nombre de las columnas entre comillas, movimos las tablas al schemas public en lugar de tener todas nuestras tablas en otro llamado librería, por lo que no hacía falta poner librería.”nombredelatabla” cada vez que nos referíamos a una de nuestras tablas.

Además, eliminamos las columnas de autor_id y libro_id de autor y libro respectivamente pasando a ser ahora el nombre y el título las primary keys (respectivamente), con lo cual no es necesario cruzar tantas tablas para conseguir los campos del nombre del autor y del título del libro, por lo que es más rápido. También movemos la columna precio de la tabla venta a la tabla vende, pues resulta más sencillo y además, de esta forma, en vez de tener por cada venta el precio total pagado, tenemos el precio de cada artículo vendido.

Estas son las consultas de la práctica anterior con sus resultados:

```
select count(isbn), count(idioma)
from edicion
where titulo = 'El Asedio'
```

Output pane

| | count bigint | count bigint |
|---|-----------------|-----------------|
| 1 | 2 | 2 |

```
select count(v.isbn)
from edicion as e, vende as v, escribe as a
where a.autor = 'Cervantes' and
      e.titulo = a.libro and
      v.isbn = e.isbn
```

Output pane

| | count bigint |
|---|-----------------|
| 1 | 260 |

```
select count(ve.isbn)
from edicion as e, vende as ve, escribe as es, aplica as a, venta as va, oferta as o
where es.autor = 'Cervantes' and
      e.titulo = es.libro and
      a.isbn = e.isbn and
      o.of_id = a.of_id and
      ve.isbn = e.isbn and
      va.venta_id = ve.venta_id and
      va.fecha > o.fecha_inicio and
      va.fecha < o.fecha_final
```

Output pane

| | count bigint |
|---|-----------------|
| 1 | 0 |

```
select sum(v.importe)
from   edicion as e, vende as v
where  e.editor = 'CreateSpace Independent Publishing Platform ' and
       e.isbn = v.isbn
```

Output pane

Data Output Explain Messages History

| | sum double precision |
|---|-------------------------|
| 1 | 11488.87 |

```
select count (ve.isbn)
from   venta as va, vende as ve
where  va.cl_id != '0' and
       ve.venta_id = va.venta_id
```

Output pane

Data Output Explain Messages History

| | count bigint |
|---|-----------------|
| 1 | 24843 |

```
select count(va.cl_id)
from   edicion as e, vende as ve, venta as va
where  e.idioma = 'Inglés' and
       ve.isbn = e.isbn and
       va.venta_id = ve.venta_id and
       va.cl_id != '0'
```

Output pane

Data Output Explain Messages History

| | count bigint |
|---|-----------------|
| 1 | 11717 |

```
select sum(v.importe)
from   edicion as e, vende as v
where  e.idioma = 'Francés' and
       v.isbn = e.isbn
```

Output pane

Data Output Explain Messages History

| | sum double precision |
|---|-------------------------|
| 1 | 26150.009999999999 |

```

select o.fecha_inicio, o.fecha_final
from oferta as o, aplica as a, edicion as e
where e.editor = 'Alfaguara ' and
      a.isbn = e.isbn and
      o.of_id = a.of_id

```

Output pane

Data Output

Explain

Messages

History

| | fecha_inicio date | fecha_final date |
|---|----------------------|---------------------|
| 1 | 2013-01-05 | 2016-12-25 |
| 2 | 2013-01-05 | 2016-12-25 |

```

select c.scrname
from (select c.cl_id as cl_id
      from cliente as c, venta as va, vende as ve, edicion as e
      where e.formato = 'Libro de bolsillo' and
            ve.isbn = e.isbn and
            va.venta_id = ve.venta_id and
            c.cl_id = va.cl_id) as t, cliente as c
where c.cl_id not in (t.cl_id) and
      c.cl_id != '0'

```

Output pane

Data Output

Explain

Messages

History

| | scrname character varying |
|----|------------------------------|
| 1 | jasonlk |
| 2 | srcasm |
| 3 | elbruno |
| 4 | gimenete |
| 5 | shashashasha |
| 6 | mattcutts |
| 7 | wilshipley |
| 8 | Gizmodo |
| 9 | danpeguine |
| 10 | neilepstein |

Programas en C

NOTA: Es necesario crear un usuario con `cl_id = 0` antes de ejecutar los programas, en concreto el `compra.c` dado que es el que sirve para identificar a los usuarios no registrados.

En lo que respecta a los programas en C que debíamos crear para interactuar con la base de datos, utilizamos la librería ODBC para ejecutar determinadas operaciones en la base de datos.

El primer programa en C es usuario. Este programa permite tanto añadir un nuevo usuario a la base de datos como eliminarlo (al eliminarlo, no se borra físicamente de la base de datos, sino que en la columna borrado, de tipo boolean, pasará a valer TRUE).

Usuario+: Primero comprobamos que no existe ningún usuario no borrado con el screen_name leído desde la terminal, calculamos el máximo cliente_id que ya existe en la base de datos, creamos una nueva tupla en cliente con el máximo id +1 y los datos que recibimos desde la terminal.

Usuario-: Actualizamos el campo borrado de la tupla correspondiente a TRUE.

```
e336673@14-20-66-223:~/Desktop/programas$ ./usuario + cabana "Alejandro Cabana"
select cl_id from cliente where scrname = 'cabana' and borrado = FALSE
Query ejecutada
insert into cliente values ('4215922', 'cabana', 'Alejandro Cabana', NULL, NULL,
    current_date, FALSE)
Usuario insertado

e336673@14-20-66-223:~/Desktop/programas$ ./usuario - cabana
update cliente set borrado = TRUE where scrname = 'cabana' and borrado = FALSE
Usuario borrado
```

El siguiente programa es oferta, que permite añadir a nuestra base de datos una nueva oferta. El descuento correspondiente a la oferta será un entero entre 0 (ningún descuento) y 100 (gratis), teniendo cada oferta un identificador único.

Al igual que con usuario, calculamos el máximo of_id que ya existe en la base de datos para luego sumarle 1 para tener una nueva oferta. Creamos una nueva oferta con los valores leídos desde la terminal, insertamos en aplica una tupla por cada isbn, relacionándolo con el of_id calculado previamente.

```
e336673@14-20-66-223:~/Desktop/programas$ ./oferta 10 2015-4-5 2016-12-12 014118
2598
insert into oferta values ('10', '1', '2015-4-5', '2016-12-12')
Oferta insertada
insert into aplica values ('1', '0141182598')
Tupla insertada
```

El siguiente programa es compra, que permite a un usuario registrado comprar todos los libros que quiera (pasados desde la terminal a través de sus isbns), donde el programa imprime en pantalla el coste de cada libro (uno por línea) teniendo en cuenta el descuento, y el coste total de la compra.

El programa, por cada isbn, obtiene los descuentos de las ofertas asociadas a él, y a través de una función que implementamos, calcula el factor por el que hay que multiplicar el precio (para aplicar los descuentos necesarios) y lo devuelve. En otra función, crea la venta con un nuevo venta_id calculado como en los programas anteriores (selecciona el máximo venta_id existente o, si no hay ninguna venta aun en la base de datos, venta_id = 0), y en aplica mete una tupla nueva por cada isbn, relacionandolo con el venta_id calculado y con el precio de ese isbn multiplicado por el factor calculado anteriormente y por otro modificador del precio que toma como valor 0,9 o 1,0 que representa el descuento por ser cliente registrado. Al final, imprime tanto los precios parciales de cada libro como el importe total.

```
e336673@14-20-66-223:~/Desktop/programas$ ./compra cabana 0141182598
select cl_id from cliente where scrname = 'cabana' and borrado = FALSE;
Screen name correcto
insert into venta values ('5102', current_date, '4215923')
Venta insertada
select precio from edicion where isbn = '0141182598'
select exp(sum(ln(1-o.descuento/100))) from oferta as o, aplica as a where o.of_
id = a.of_id and a.isbn = '0141182598';
Query ejecutada
insert into vende values ('5102', '0141182598', '7.49')
Tupla insertada
0141182598: 7.49€
Importe total: 7.49€
```

El último programa a implementar es best_seller, programa que muestra los libros más vendidos. El programa lee desde la terminal el número de libros más vendidos a mostrar e imprime por pantalla el título y el número de copias vendidas de los mismos.

Este programa, a través de una consulta, generamos una tabla con cada título y el número de copias vendidas del mismo, ordenadas de forma descendente y cogemos las n primeras filas de la tabla (siendo n un número leído desde la terminal).

```
e336673@14-20-66-223:~/Desktop/programas$ ./best_seller 5
1. Don Quixote: 180
2. Cervantes: 105
3. El caballero de Olmedo: 90
4. El Asedio: 88
5. Chomsky Para Principiantes: 86
```