

PRÁCTICAS DE ESTRUCTURAS DE DATOS

MEMORIA PRÁCTICA 2

David Álvarez Sáez

Grupo 1262

Índice

Introducción y menú principal	2
Submenú products	3
Consulta products stock	3
Consulta find	3
Submenú orders	5
Consulta open	5
Consulta range	6
Consulta detail	7
Submenú customers	9
Consulta find	9
Consulta list products	10
Consulta balance	11
Splint	13

Introducción y menú principal

En esta práctica no se alcanzan los datos de la base de datos por medio de las consultas directamente, sino que se implementan unas funciones en c para que el acceso sea mucho más fácil y de esta manera el usuario no requiera de conocimientos sobre sql.

En primer lugar se encuentra un menú en el que se ha de seleccionar el submenú que esté relacionado con la consulta que se quiera realizar. Estos datos son sobre los productos (“products”), pedidos (“orders”) y clientes (“customers”). Seleccionando cada uno se podrá acceder a las consultas concretas de cada submenú. En caso de querer salir del programa sin realizar ninguna consulta podemos encontrar una opción para ello, “Back”.

Para la implementación del programa se han creado varios ficheros. Por un lado, el principal, que es menu.c, que se encarga de mostrar el menú donde se eligen las diferentes opciones y realiza llamadas a las funciones que muestran los submenús y a las que permiten interactuar con ellos. Estas últimas se encuentran en los ficheros products.c, orders.c y customers.c, habiendo una función para cada consulta disponible.

Submenú products

Consulta products stock

Esta consulta muestra la cantidad de un producto en el almacén tras haber indicado el código del producto en cuestión. La implementación en sql es:

```
SELECT p.quantityinstock
FROM   products p
WHERE  p.productcode = ?;
```

La ejecución de products_stock.sh es:

```
eps@labvirteps:~/Escritorio/p2edat$ ./products_stock.sh
spawn ./menu
This is where you can find all the information about the products, orders and customers
(1) Products
(2) Orders
(3) Customers
(4) Quit

Enter a number that corresponds to your choice > 1

(1) Stock
(2) Find
(3) Back

Enter a number that corresponds to your choice > 1

Enter productcode > S10_1678
7933
Enter productcode >
-----OK
```

Consulta find

La consulta find nos permite saber cuál es el código del producto con haber introducido el nombre de éste, no es necesario que esté completo. La implementación en sql es:

```
SELECT p.productcode,
       p.productname
FROM   products p
WHERE  p.productname LIKE '%%%s%%'
ORDER BY p.productcode;
```

La ejecución de products_find.sh es:

```
eps@labvirteps:~/Escritorio/p2edat$ ./products_find.sh
spawn ./menu
This is where you can find all the information about the products, orders and customers
(1) Products
(2) Orders
(3) Customers
(4) Quit

Enter a number that corresponds to your choice > 1

(1) Stock
(2) Find
(3) Back

Enter a number that corresponds to your choice > 2

Enter productname > Harley
S10_1678
S10_4698
S18_2625
Enter productname >
-----OK
_
```

Submenú orders

Consulta open

En esta consulta se muestran los números de los pedidos que todavía no se han enviado. La idea para la implementación de esta consulta es seleccionar los productos cuyo campo de entrega esté vacío (sea NULL), es decir, todavía no entregado. La implementación en sql es:

```
SELECT o.ordernumber
FROM   orders o
WHERE  o.shippeddate IS NULL;
```

La ejecución de orders_open.sh es:

```
^Ceps@labvirtips:~/Escritorio/p2edat$ ./orders_open.sh
spawn ./menu
This is where you can find all the information about the products, orders and customers
(1) Products
(2) Orders
(3) Customers
(4) Quit

Enter a number that corresponds to your choice > 2

(1) Open
(2) Range
(3) Detail
(4) Back

Enter a number that corresponds to your choice > 1

10167
10248
10260
10262
10334
10401
10407
10414
10420
10421
10422
10423
10424
10425

-----OK
```

Consulta range

En esta consulta se muestran el número de cada pedido junto con el día en el que fue pedido y el día que se ha entregado. La idea para la implementación de esta consulta es restringir las tuplas solamente a las que cumplan la condición de estar entre ciertas dos fechas y de la tabla resultante se eligen los datos especificados. La implementación en sql es:

```
SELECT o.ordernumber,
       o.orderdate,
       o.shippeddate
FROM   orders o
WHERE  o.orderdate >= ?
      AND o.orderdate <= ?
ORDER BY o.ordernumber;
```

La ejecución de orders_open.sh es:

```
eps@labvirtips:~/Escritorio/p2edat$ ./orders_range.sh
spawn ./menu
This is where you can find all the information about the products, orders and customers
(1) Products
(2) Orders
(3) Customers
(4) Quit

Enter a number that corresponds to your choice > 2

(1) Open
(2) Range
(3) Detail
(4) Back

Enter a number that corresponds to your choice > 2

Enter dates (YYYY-MM-DD - YYYY-MM-DD) > 2003-01-10 - 2003-04-21
10102 2003-01-10 2003-01-14
10103 2003-01-29 2003-02-02
10104 2003-01-31 2003-02-01
10105 2003-02-11 2003-02-12
10106 2003-02-17 2003-02-21
10107 2003-02-24 2003-02-26
10108 2003-03-03 2003-03-08
10109 2003-03-10 2003-03-11
10110 2003-03-18 2003-03-20
10111 2003-03-25 2003-03-30
10112 2003-03-24 2003-03-29
10113 2003-03-26 2003-03-27
10114 2003-04-01 2003-04-02
10115 2003-04-04 2003-04-07
10116 2003-04-11 2003-04-13
10117 2003-04-16 2003-04-17
10118 2003-04-21 2003-04-26
Enter dates (YYYY-MM-DD - YYYY-MM-DD) >
-----OK
```

Consulta detail

En esta consulta se muestran los detalles sobre un pedido en concreto. Para implementar esta consulta he hecho uso de dos queries distintas: por un lado, la primera devuelve el precio total, el estado del envío (si ya ha sido entregado o no), y la fecha en la que fue pedido; por otro lado, la segunda consulta se encarga de devolver el código del producto junto a la cantidad pedida y el precio de cada unidad. Las implementaciones en sql son:

```
SELECT o.orderdate,  
       o.status,  
       Sum(ol.quantityordered * ol.priceeach)  
FROM   orders o  
       JOIN orderdetails ol  
         ON o.ordernumber = ol.ordernumber  
WHERE  o.ordernumber = ?  
GROUP BY o.orderdate,  
         o.status;
```

```
SELECT ol.productcode,  
       ol.quantityordered,  
       ol.priceeach  
FROM   orders o  
       JOIN orderdetails ol  
         ON o.ordernumber = ol.ordernumber  
WHERE  o.ordernumber = ?  
ORDER BY ol.orderlinenumber;
```


La ejecución de orders_detail.sh es:

```
eps@labvirteps:~/Escritorio/p2edat$ ./orders_detail.sh
spawn ./menu
This is where you can find all the information about the products, orders and customers
(1) Products
(2) Orders
(3) Customers
(4) Quit

Enter a number that corresponds to your choice > 2

(1) Open
(2) Range
(3) Detail
(4) Back

Enter a number that corresponds to your choice > 3

Enter ordernumber > 10100
Order Date=2003-01-06 - status=Shipped
Total sum = 10223.830000
S24_3969 49 35.29
S18_2248 50 55.09
S18_1749 30 136.00
S18_4409 22 75.46
Enter ordernumber >
-----OK
```

Submenú customers

Consulta find

En esta consulta se muestra el nombre del contacto y su apellido, el nombre del cliente y el número del pedido. Para la implementación de esta query se selecciona solamente los nombres y apellidos del contacto pedido y se seleccionan las tuplas concretas. La implementación en sql es:

```
SELECT c.customernumber,  
       c.customername,  
       c.contactfirstname,  
       c.contactlastname  
FROM   customers c  
WHERE  ( c.contactfirstname LIKE '%s%%' )  
       OR ( c.contactlastname LIKE '%s%%' )  
ORDER BY c.customernumber;
```

La ejecución de customers_find.sh es:

```
eps@labvirteps:~/Escritorio/p2edat$ ./customers_find.sh  
spawn ./menu  
This is where you can find all the information about the products, orders and customers  
(1) Products  
(2) Orders  
(3) Customers  
(4) Quit  
  
Enter a number that corresponds to your choice > 3  
  
(1) Find  
(2) List Products  
(3) Balance  
(4) Back  
  
Enter a number that corresponds to your choice > 1  
  
Enter customer name > Mary  
146 Saveley & Henriot, Co. Mary Saveley  
219 Boards & Toys Co. Mary Young  
Enter customer name >  
-----OK
```

Consulta list products

En esta consulta se muestra el nombre de un producto y todas las unidades solicitadas por el cliente que introduce el usuario. La idea para la implementación es mostrar el nombre de cada producto e ir recorriendo la tabla con los detalles del pedido, sumando las cantidades de cada producto para mostrarlo como resultado final. La implementación en sql es:

```
SELECT p.productname,  
       Sum(ol.quantityordered) AS result  
FROM   customers c,  
       products p,  
       orderdetails ol,  
       orders o  
WHERE  c.customernumber = o.customernumber  
AND    c.customernumber = ?  
AND    ol.productcode = p.productcode  
AND    o.ordernumber = ol.ordernumber  
GROUP BY p.productname,  
         p.productcode  
ORDER BY p.productcode
```

La ejecución de customers_list_products.sh es:

```
eps@labvirtips:~/Escritorio/p2edat$ ./customers_list_products.sh  
spawn ./menu  
This is where you can find all the information about the products, orders and customers  
(1) Products  
(2) Orders  
(3) Customers  
(4) Quit  
  
Enter a number that corresponds to your choice > 3  
  
(1) Find  
(2) List Products  
(3) Balance  
(4) Back  
  
Enter a number that corresponds to your choice > 2  
  
Enter customer number > 141  
1969 Harley Davidson Ultimate Chopper 66  
1952 Alpine Renault 1300 50  
1996 Moto Guzzi 1100i 45  
2003 Harley-Davidson Eagle Drag Bike 56  
1972 Alfa Romeo GTA 26  
1968 Ford Mustang 20  
2001 Ferrari Enzo 20  
1958 Setra Bus 49  
2002 Suzuki XREO 21  
1969 Corvair Monza 125  
1968 Dodge Charger 59  
1969 Ford Falcon 49  
1970 Plymouth Hemi Cuda 44  
1957 Chevy Pickup 183  
1969 Dodge Charger 80  
1940 Ford Pickup Truck 54  
1993 Mazda RX-7 29  
1937 Lincoln Berline 103  
1936 Mercedes-Benz 500K Special Roadster 68  
1965 Aston Martin DB5 101  
1980s Black Hawk Helicopter 74  
1917 Grand Touring Sedan 39  
1948 Porsche 356-A Roadster 65  
1995 Honda Civic 41
```

```

1912 Ford Model T Delivery Wagon 30
1969 Chevrolet Camaro Z28 129
1971 Alpine Renault 1600s 124
1937 Horch 930V Limousine 81
2002 Chevy Corvette 174
1940 Ford Delivery Sedan 74
1956 Porsche 356A Coupe 161
Corsair F4U ( Bird Cage) 89
1936 Mercedes Benz 500k Roadster 43
1992 Porsche Cayenne Turbo Silver 161
1936 Chrysler Airflow 81
1900s Vintage Tri-Plane 96
1961 Chevrolet Impala 69
1980s GM Manhattan Express 41
1997 BMW F650 ST 23
1982 Ducati 996 R 124
1954 Greyhound Scenicruiser 157
1950's Chicago Surface Lines Streetcar 150
1996 Peterbilt 379 Stake Bed with Outrigger 80
1928 Ford Phaeton Deluxe 89
1974 Ducati 350 Mk3 Desmo 82
1930 Buick Marquette Phaeton 96
Diamond T620 Semi-Skirted Tanker 141
1962 City of Detroit Streetcar 145
2002 Yamaha YZR M1 40
The Schooner Bluenose 89
American Airlines: B767-300 112
The Mayflower 91
HMS Bounty 73
America West Airlines B757-200 70
The USS Constitution Ship 66
1982 Camaro Z28 73
ATA: B757-300 27
F/A 18 Hornet 1/72 59
The Titanic 126
The Queen Mary 75
American Airlines: MD-11S 79
Boeing X-32A JSF 71
Pont Yacht 113
Enter customer number >
-----OK

```

Consulta balance

En esta consulta se compara lo ya pagado con lo que está por pagar respecto al cliente introducido, utilizando el número asociado a cada cliente. La implementación en sql es:

```

SELECT ( payments - products ) AS balance
FROM (SELECT c1.customernumber,
            Sum(p.amount) AS payments
      FROM customers c1,
           payments p
     WHERE c1.customernumber = p.customernumber
    GROUP BY c1.customernumber)AS p1
JOIN (SELECT c2.customernumber,
            Sum(o1.priceeach * o1.quantityordered) AS products
      FROM customers c2,
           orders o,
           orderdetails o1
     WHERE c2.customernumber = ?
           AND c2.customernumber = o.customernumber
           AND o.ordernumber = o1.ordernumber
    GROUP BY c2.customernumber)AS p2
ON p1.customernumber = p2.customernumber;

```

La ejecución de customers_balance.sh es:

```
eps@labvirteps:~/Escritorio/p2edat$ ./customers_balance.sh
spawn ./menu
This is where you can find all the information about the products, orders and customers
(1) Products
(2) Orders
(3) Customers
(4) Quit

Enter a number that corresponds to your choice > 3

(1) Find
(2) List Products
(3) Balance
(4) Back

Enter a number that corresponds to your choice > 3

Enter customer number > 141
Balance = -104950.56
Enter customer number >
-----OK
```

Splint

Algunos de los fallos observables al haber ejecutado “splint -nullpass *.c *.h” son, por un lado, respecto a la función `sprintf`, nos indica que deberíamos sustituirlo por otro semejante, este es `snprintf`. El `snprintf` es similar al `sprintf` pero evita el posible overflow del buffer, para esto basta con añadir un parámetro más que será la longitud máxima de la cadena. Este error no está corregido porque al sustituir `sprintf` por `snprintf`, fijandome en añadir este parámetro, al compilar daba warning dando a entender que quería decir `sprintf`, habiendome asegurado de haber incluido `<stdio.h>`.

El arreglo en el caso del `products.c` sirve con: `(void) snprintf(query, (size_t) (BufferLength + 108), “SELECT p.productcode, p.productcode, FROM products p WHERE p.productname LIKE ‘%%s%%’ ORDER BY p.productcode;”, productname);`

De manera similar se solucionaría en el `customers.c`.

Respecto a los otros code warnings sobre el `menu.c`, estos aparecen por haber separado el código en diferentes funciones para no hacer ilegible el `menu.c` pero para subsanar estos errores solamente hay que colocar el código de cada función en el lugar que es llamada ésta.


```

eps@labvirteps:~/Escritorio/p2edat$ splint -nullpass *.c *.h
Splint 3.1.2 --- 20 Feb 2018

customers.c: (in function customers_find)
customers.c:41:9: Buffer overflow possible with sprintf. Recommend using
      snprintf instead: sprintf
  Use of function that may lead to buffer overflow. (Use -bufferoverflowhigh to
  inhibit warning)
products.c: (in function products_find)
products.c:103:16: Buffer overflow possible with sprintf. Recommend using
      snprintf instead: sprintf
menu.c:23:5: Function exported but not used outside menu: ShowMainMenu
  A declaration is exported, but not used outside this module. Declaration can
  use static qualifier. (Use -exportlocal to inhibit warning)
  menu.c:109:1: Definition of ShowMainMenu
menu.c:24:6: Function exported but not used outside menu: ShowProductsMenu
  menu.c:142:1: Definition of ShowProductsMenu
menu.c:25:5: Function exported but not used outside menu: ShowProductsSubMenu
  menu.c:177:1: Definition of ShowProductsSubMenu
menu.c:26:6: Function exported but not used outside menu: ShowOrdersMenu
  menu.c:216:1: Definition of ShowOrdersMenu
menu.c:27:5: Function exported but not used outside menu: ShowOrdersSubMenu
  menu.c:250:1: Definition of ShowOrdersSubMenu
menu.c:28:6: Function exported but not used outside menu: ShowCustomersMenu
  menu.c:290:1: Definition of ShowCustomersMenu
menu.c:29:5: Function exported but not used outside menu: ShowCustomersSubMenu
  menu.c:324:1: Definition of ShowCustomersSubMenu

Finished checking --- 9 code warnings_

```