

Práctica 2

CAMILO JENÉ

02-01-2020

—

ESTRUCTURAS DE DATOS

—

G - 1261

INSTRUCCIONES

En esta práctica, vamos a comenzar a tratar con el uso de interfaces de búsqueda. El usuario nunca manda la instrucción directamente y eso es lo que vamos a implementar mediante un menú y la conexión de c con SQL.



EL PROCESO

Como ya he comentado en la intro, vamos a hacer uso del lenguaje C para ejecutar una serie de operaciones que ayuden al usuario a realizar diversas consultas. Para ello, la librería usada será ODBC.

Nos requieren realizar un menú en el que el usuario se moverá y cuando le pida un dato por teclado, éste le devuelva la consulta deseada.

A su vez, debemos realizar la funcionalidad del programa. Cada apartado del menú lo involucraremos en un objetivo concreto. Estos objetivos se dividirán en tres apartados: products, customers y orders.

PRODUCTS

En este apartado realizaremos dos métodos:

- **showStock():** Este método se encargará de devolver el stock de un producto pasado por teclado. Para ello, recibiremos ese valor, y mediante un **snprintf**, introduciremos la consulta (con ese valor) dentro de una variable que usaremos para enviar mediante **SQLExecDirect** la información necesaria para realizar la búsqueda. La variable **stmt** será la que guardará toda la información necesaria. La consulta es la siguiente:

```
SELECT quantityinstock
FROM products
WHERE productcode = '%s'
```

- **showFind():** La función se encargará de pedir por teclado el nombre del producto y devolverá un listado con todos los productos que contengan ese nombre. Por ello, pongo %% delante y detrás de la cadena introducida por teclado, para así indicar en SQL que da igual lo que haya delante y detrás del nombre. La consulta exacta es la siguiente:

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

ORDERS

En este apartado nos piden introducir 3 funciones para el tratado de pedidos. Estas son:

- **showOpen():** Función que muestra por pantalla el listado de pedidos que aún no se han enviado. La consulta es la siguiente:

```
SELECT ordernumber
FROM orders
WHERE shippeddate IS NULL
ORDER BY ordernumber
```

- **showRange():** Esta función se va a encargar de mostrarnos por pantalla un listado de todos los pedidos solicitados entre dos fechas que nos va a pasar por teclado el usuario. Hay que tener en cuenta que para ello vamos a anidar una serie de mayores e iguales ya que deseamos buscar los que se encuentren entre ambos rangos. La consulta sería la siguiente:

```
SELECT ordernumber,
       orderdate,
       shippeddate
FROM orders
WHERE ( orderdate <= '%s'
       AND orderdate >= '%s' )
       OR ( orderdate >= '%s'
       AND orderdate <= '%s' )
ORDER BY ordernumber
```

- **showDetail():** Por último en este apartado tenemos la función que realizará una de las funciones más complejas de la práctica. Para esta función, vamos a realizar 3 consultas en una. Para ello debemos conectarnos y desconectarnos de la base de datos hasta 2 veces. Primero realizamos la consulta de la fecha en la que se realizó el pedido:

```
SELECT orderdate,
       status
FROM orders
WHERE ordernumber = '%d'
```

A continuación, realizamos la consulta del coste total del pedido:

```
SELECT Sum(quantityordered * priceeach)
FROM orderdetails
WHERE ordernumber = '%d'
```

Por último, mostraremos por pantalla el id, la cantidad de unidades solicitadas y el precio por unidad:

```
SELECT productcode,
        quantityordered,
        priceeach
FROM    orderdetails
WHERE   ordernumber = '%d'
ORDER BY orderlinenumber
```

CUSTOMERS

En este apartado nos piden realizar consultas sobre la información de los clientes. He realizado las siguientes funciones:

- **showFindCustomer():** Pediremos una cadena por teclado al usuario y mostraremos todos aquellos clientes que contengan en su apellido o en su nombre la cadena pasada. La consulta es la siguiente:

```
SELECT customernumber,
        customername,
        contactfirstname,
        contactlastname
FROM    customers
WHERE   contactfirstname LIKE '%%%s%%'
        OR contactlastname LIKE '%%%s%%'
```

- **showProducts():** Pediremos un cliente por teclado y devolveremos un listado de todos los productos pedidos por ese cliente. Lo importante de esta consulta es que debemos mostrar el numero total de unidades solicitadas. Por ello, utilizamos la función SUM. La consulta quedaría de la siguiente manera:

```
SELECT p.productname,
        Sum(o2.quantityordered) AS suma
FROM    orders AS o1,
        orderdetails AS o2,
        products AS p
WHERE   o1.ordernumber = o2.ordernumber
        AND o2.productcode = p.productcode
        AND o1.customernumber = '%d'
GROUP BY p.productname,
        p.productcode
ORDER BY p.productcode
```

- **showBalance():** Por último, se nos pide devolver el saldo de un cliente pasado por teclado. Para ello, debemos utilizar la función SUM, que se encargará de sumar los pagos realizados menos la suma de los productos comprados, es decir, debemos realizar una suma anidada. La consulta es la siguiente:


```
SELECT (( Sum(p.amount) ) - ( Sum(o2.priceeach * o2.quantityordered) ) ) AS BALANCE
FROM   payments AS p,
       orders AS o1,
       orderdetails AS o2
WHERE  o1.ordernumber = o2.ordernumber
AND    p.customernumber = o1.customernumber
AND    o1.customernumber = '%d'
```

OBSERVACIONES

Para terminar, me gustaría comentar una serie de observaciones. Durante toda la práctica, utilizo las mismas funciones. Éstas se encargan según mi entendimiento de:

- **odbc_connect:** conectar a la base de datos.
- **SQLAllocHandle:** asignar espacio para la declaración que se encargará de manejar el puntero stmt.
- **SQLExecDirect:** enviar el puntero con la consulta a la base de datos.
- **SQLNumResultCols:** devolver el numero de columnas de la consulta.
- **SQLDescribeCol:** describir el tipo de columna.
- **SQLFetch:** escribir el dato recibido de la consulta.
- **odbc_disconnect:** desconectar de la base de datos,

Se nos pide pasar el splint. Como no me da ningún error, el fichero splint.log entregado está vacío. Si hubiese encontrado algún error este se habría impreso en el log.

