

Escuela Politécnica Superior, UAM

Estructura de Datos 2016-2017

Práctica 2. Bases de datos y ODBC.

Fecha límite de entrega de la práctica: En **la semana del 13 de Noviembre**.

Cada grupo entregará la práctica el día de su clase, 2 horas antes del comienzo de clase.

La entrega de la práctica se realizará a través de Moodle. El retraso en la entrega se penalizará con 1 punto por cada día natural después de la fecha límite.

Consejos útiles

1. Las memorias, cuando se piden, son importantes: no sólo sirven para que vuestro profesor tenga una idea de lo que habéis hecho y cómo lo habéis organizado; también proporciona indicaciones importantes sobre el razonamiento que habéis seguido para hacer las cosas de una determinada manera. Esto nos ayudará a evaluar mejor vuestro trabajo. Además, escribir una buena memoria es importante para los que, como vosotros, trabajarán en campo técnico: hacer que los colegas, los jefes y los clientes entiendan vuestro trabajo será una de vuestras actividades más importantes.
2. Puede parecer banal, pero... ¡poned el nombre de los componentes del grupo en TODOS los ficheros! No os podéis ni siquiera imaginar cuantas maneras hay de perder el rastro de quien ha hecho algo.

Objetivos

El usuario final no interactúa directamente con una base de datos, si no que lo hace a través de un programa que construye las consultas y las lanza contra la base de datos. Esto es lo que ocurre, por ejemplo, cada vez que accedemos a una página web en un sitio de cierta envergadura que contiene muchos datos.

En esta práctica experimentaremos con el acceso a base de datos desde programas en C. En particular, usaremos la librería ODBC para crear una serie de programas que actuarán sobre la base de datos de la librería que hemos diseñado en la primera práctica.

Entrega

La entrega se realizará mediante un único fichero en formato .tgz o .zip que incluirá una estructura con los siguientes directorios: “**basedatos**”, “**consultas**” y “**odbc**”. Cada uno de estos directorios incluirá el contenido requerido en la parte de la práctica correspondiente.

El nombre del fichero tendrá el formato EDAT1617_p2_XXX_AAA_BBB.zip donde:

- XX es el número de la pareja de prácticas
- AAA y BBB son los apellidos de los miembros de la pareja.

Este fichero se entregará mediante moodle, usando el enlace que se publicará en la página de la asignatura. Sólo tendrá que entregar el fichero UNO de los miembros de la pareja

Parte I: Llenado de la base de datos

Vamos a insertar los datos en las tablas. Aquí tendremos que resolver un problema muy común a la hora de poblar bases de datos con datos existentes: el esquema de los datos tal y como aparece en el fichero no se corresponde con las tablas que hemos definido en el diseño relacional. Será necesario “convencer” los datos para que se metan en nuestra tabla. Esta parte será claramente diferente por cada pareja, dado que cada pareja tendrá su diseño individual con tablas ligeramente distintas. Las bases son, por otro lado, iguales para todos, y las podemos dividir en cinco pasos:

1. se crea una colección de tablas temporales, una tabla por cada fichero de datos. Cada tabla tendrá un número de columnas igual al número de campos del fichero y de los tipos adecuados para recoger los datos del fichero;
2. mediante la instrucción COPY se insertan los datos de los ficheros en las tablas temporales;
3. se crean las tablas del diseño relacional elegido;
4. se transforman los datos: por cada tabla del diseño relacional se crea una consulta que, partiendo de los datos de las tablas temporales, crea la tabla del diseño relacional; los resultados de la consulta se insertan físicamente en la tabla usando la consulta dentro de un INSERT INTO;

5. con un DROP TABLE se borran las tablas temporales.

Tarea

- a. Crear una base de datos “libros” en PostgreSQL que implemente el modelo relacional de la práctica 1 con las modificaciones necesarias que la hagan coherente con los datos a cargar.
- b. Cargar las tablas con los ficheros de datos proporcionados para esta práctica y que están disponibles en Moodle.

NOTA: La base de datos debe llamarse “libros”

Entrega

- Fichero con el volcado de la base de datos. Utilizar “pgdump”.
- Fichero “adaptación.txt” donde se describan las modificaciones incluidas en la implementación de la base de datos respecto al modelo relacional
- Fichero “llenado.txt” que incluya:
 - o Una sección “adaptación” donde se describan las modificaciones incluidas en la implementación de la base de datos respecto al modelo relacional presentado en la práctica 1.
 - o Un listado detallado de todos los comandos en orden de ejecución y tablas temporales utilizados para el llenado de las tablas finales.

Parte II. Consultas

Tarea

- Implementar mediante SQL de PostgreSQL las consultas propuestas en la práctica 1.
- Muy recomendable utilizar una herramienta tipo pgAdmin que permite por ejemplo depurar partes de la consulta o guardar en un fichero texto las consultas que se van escribiendo.

Entrega

- Un fichero “sql_consultas.txt” con la codificación SQL de las consultas.
- Una serie de ficheros “resultado_1.txt”, “resultado_2.txt”, que corresponden al resultado de la ejecución de cada una de las consultas.

Parte III. Acceso a base de datos desde C

En este apartado usaremos ODBC para crear unos programas en C que ejecuten ciertas operaciones en la base de datos. ODBC es una librería para enviar consultas a una base de datos y recibir resultados. En clase veremos los conceptos fundamentales de ODBC y como usarla para interactuar con la base de datos.

El enlace recomendado con información sobre ODBC: La sección “manuals” de <http://www.unixodbc.org>. Y dentro de esta sección, muy recomendable:

- “[Programming Manual tutorial](#)”, para comenzar paso a paso con ODBC, desde primera compilación pasando por conexión con la base de datos, hasta recuperación de resultados de consultas.
- “[ODBC from C Tutorial Part 1](#)”, con explicaciones y ejemplos en pequeños trozos de código de cada una de las fases de ODBC. Completa muy bien el manual anterior.

Se crearán los programas siguientes:

oferta

Este programa permite la introducción de una oferta. Para añadir una oferta se ejecutará:

oferta <descuento> <desde> <hasta> <isbn> <isbn> <isbn>

donde descuento es un entero entre 0 (ningún descuento) y 100 (regalamos el libro). La lista de isbn tiene longitud arbitraria, y representa todos los libros a que se aplica el descuento.

compra

Con este programa se podrá, bien añadir una venta, bien borrar todas las ventas asociadas a una factura. Para esto se ejecutará:

compra add <screen_name> <fecha> <isbn> <isbn> <isbn>

compra del <factura_id>

Nota: Todas las compras de un usuario en una misma fecha se englobarán en una misma

factura. Si es la primera compra de un usuario en un día, se generará un nuevo código de factura.

llena_ventas

Este programa sirve para llenar la tabla “ventas” a partir de un fichero de entrada cuyo formato se corresponde con el fichero “ventas.txt”.

llena_ventas <fichero de entrada>

Realiza una comparación entre los datos de la tabla “ventas” resultantes de su llenado en la parte 2 de la práctica y los datos de esta misma tabla con el llenado mediante el programa “llena_ventas”: ¿mismo número de entradas? Si no, ¿por qué’?

Tarea

- Implementar los programas antes descritos en C y ODBC.
- No hay que realizar control de datos de entrada. Se supone que están bien en cuanto a formato.
- En todas las llamadas ODBC se deberá realizar control de errores y se imprimirá el error detectado.
- La conexión se realizará la base de datos “libros” y como credenciales de conexión se utilizará usuario: “alumnodb” y password: “alumnodb”
- El código debe de estar depurado con valgrind.

Entrega

- Códigos fuente de los programas.
- Makefile necesario para su compilación.