

Proyecto #1 DBMS - TRIOS

I. Modalidad y fecha de entrega

- a) El proyecto debe hacerse en grupos de tres personas y ser enviado antes de la fecha límite de entrega:
Martes 04 de abril de 2017 a las 6:10pm
- b) Luego de la fecha límite por cada hora tarde se restarán 10 puntos.

II. Descripción actividad

El proyecto consiste en la realización de un manejador de base de datos con la finalidad de que el estudiante comprenda el funcionamiento interno de las bases de datos y consolide los conocimientos sobre instrucciones DDL y DML de SQL. Se puede utilizar de referencia los siguientes enlaces:

- Gramática SQL-92: <http://savage.net.au/SQL/sql-92.bnf.html>
- Comandos SQL PostgreSQL: <http://www.postgresql.org/docs/9.2/interactive/sql-commands.html>
- Terminal interactiva y meta-instrucciones: <http://www.postgresql.org/docs/9.2/interactive/app-psql.html>
- Revisión de restricciones: <http://www.postgresql.org/docs/9.2/interactive/ddl-constraints.html>

EJERCICIO #1: DDL

Instrucciones a implementar:

- i) CREATE DATABASE nombre – Crea una nueva base de datos y un nuevo directorio.
- ii) ALTER DATABASE nombre RENAME TO nuevoNombre – Cambia el nombre de una base de datos. Si la base de datos se encuentra en uso, se hace el cambio de nombre y la base de datos actual sería el nuevoNombre.
- iii) DROP DATABASE nombre – Borra una base de datos.
- iv) SHOW DATABASES – Muestra las bases de datos actuales.
- v) USE DATABASE nombre – Selecciona la base de datos a trabajar
- vi) CREATE TABLE – Crea una nueva tabla y un nuevo archivo. Se realiza bajo la siguiente sintaxis.
 - (1) CREATE TABLE nombre (columna Tipo [...] CONSTRAINT C [...])
 - (2) C puede ser:
 - (a) PK_nombre PRIMARY KEY (id [...])
 - (b) FK_nombre FOREIGN KEY (id [...]) REFERENCES tabla (id [...])
 - (c) CH_nombre CHECK (Exp)
 - (3) Tipo puede ser:
 - (a) INT
 - (b) FLOAT
 - (c) DATE – Valores ingresados en este tipo deben ser fechas válidas (Formato: 'AAAA-MM-DD')
 - (d) CHAR (cantidad)
 - (4) Exp puede ser una expresión booleana que puede ser formateada por los siguientes operadores:
 - (a) Lógicos: AND, OR, NOT
 - (b) Relacionales: < , <= , > , >= , <> , =

- vii) ALTER TABLE nombre RENAME TO nuevoNombre – Cambia el nombre a una tabla
- viii) ALTER TABLE nombre acción [...]
 - (1) Donde acción puede ser:
 - (a) ADD COLUMN columna tipo CONSTRAINT C [...]
 - (b) ADD CONSTRAINT C
 - (c) DROP COLUMN columna
 - (d) DROP CONSTRAINT nombre_constraint
- ix) DROP TABLE nombre – Borra una tabla
- x) SHOW TABLES – Muestra las tablas de la base de datos actual, si esta en uso.
- xi) SHOW COLUMNS FROM nombre – Muestra la descripción de las columnas de una tabla de la base de datos actual (incluyendo las restricciones)

Observaciones a tomar en cuenta:

- i) Debe existir un archivo maestro con la meta data de las bases de datos que incluya el nombre de las bases de datos y la cantidad de tablas por cada una.
- ii) Se debe crear un directorio por cada base de datos
- iii) Dentro de cada directorio debe existir un archivo maestro con la meta data de las tablas que incluya los nombres de las tablas, cantidad de registros, restricciones, etc.
- iv) Se debe crear un archivo por cada tabla
- v) Si existe algún error, ya sea de sintaxis de SQL o de acciones en la base de datos (EJ: no se pudo borrar una tabla), el DBMS debe mostrar un mensaje de error descriptivo e interrumpir su ejecución. Esto es necesario para asegurar que la información y el esquema almacenados no se corrompan. Ya que no se está implementando la funcionalidad de “rollback” esta sería la forma más segura de proceder.
 - a. No se puede alterar, borrar, mostrar o utilizar una base de datos que no existe.
 - b. No se pueden hacer operaciones de tablas sin haber seleccionado una base de datos.
 - c. No se puede alterar, borrar, mostrar o utilizar una tabla que no existe.
- vi) Revisiones de restricciones para DROP TABLE. Si existe una referencia HACIA la tabla que se desea borrar, no se debe permitir la acción. De esta manera se garantiza la integridad referencial. El usuario del DBMS deberá primero hacer ALTER TABLE tabla DROP CONSTRAINT constraint, para luego poder borrar la tabla que está siendo referenciada.
- vii) Verificación para DROP DATABASE. Para esta instrucción se debe hacer una doble verificación con el usuario del DBMS mostrando el siguiente mensaje: “¿Borrar base de datos nombre_BD con N registros? (si/no)” Donde N es la sumatoria de los registros de todas las tablas en la base de datos.
- viii) El DBMS debe tener una opción de *verbose* (-v) al momento de comenzar para ir mostrando cada instrucción ejecutada a modo de debug. Si la opción no se encuentra seleccionada únicamente muestra los mensajes explícitos en pantalla o mensajes de error.

EJERCICIO #2: DML

Instrucciones a implementar:

- i) INSERT INTO nombre (columna [...]) VALUES (valor [...])
 - (1) Las columnas pueden ser listadas en cualquier orden
 - (a) Si no se provee una lista de columnas se toman todas las columnas de la tabla en el orden que fueron declaradas de manera predeterminada.
 - (b) Si únicamente N columnas son suplidas en la cláusula VALUES se tomarán los N primeros nombres de columnas.

- (c) Los valores suplidos por la cláusula VALUES asociados con la lista explícita o implícita de columnas es de izquierda a derecha.
- (2) Cada columna no presente en la lista explícita o implícita de columnas será llenada con el valor predeterminado o NULL.
- (3) Si la expresión para cualquier columna no es del tipo de dato correcto se debe intentar realizar una conversión automática de tipos.
- (4) Como salida normal, al finalizar exitosamente un INSERT se muestra en pantalla lo siguiente:
 - (a) "INSERT (cont) con éxito" donde *cont* es la cantidad de filas insertadas
- (5) Formato de valores de los distintos tipos de datos:
 - (a) INT – número entero base 10
 - (i) EJ: -120000, -1, 0, 15, 1000
 - (ii) Si se ingresa un real se truncan los decimales
 - (b) FLOAT – número real base 10, único carácter adicional es el punto decimal.
 - (i) EJ: -0.4236, 0.0, 452.53
 - (ii) Si se ingresa un entero se convierte en real.
 - (c) DATE – representa una fecha. Delimitado por apóstrofes. Formato: ' AAAA-MM-DD '
 - (i) EJ: '2010-04-20', '2014-02-28'
 - (ii) Verificar que la fecha sea válida
 - (d) CHAR (cantidad) – Delimitado por apóstrofes. Cantidad define la cantidad de caracteres.
 - (i) EJ: 'Este es un texto'
 - (ii) Verificar que la longitud del texto sea menor o igual a la definida para el campo.
- ii) UPDATE nombre SET columna = valor [...] WHERE condición
 - (1) Se actualizan los valores de las columnas especificadas en todas las filas que satisfagan la condición.
 - (2) Si la cláusula WHERE está ausente, el efecto es modificar TODAS las filas de la tabla.
 - (3) Únicamente las columnas a ser modificadas deben ser mencionadas en la cláusula SET.
 - (4) Las columnas que no sean explícitamente modificadas retendrán sus valores previos.
 - (5) Como salida, al terminar exitosamente un UPDATE se muestra en pantalla lo siguiente:
 - (a) "UPDATE (cont) con éxito" donde *cont* es la cantidad de filas modificadas.
- iii) DELETE FROM nombre WHERE condición
 - (1) Se eliminan las filas que satisfagan la cláusula WHERE de la tabla especificada.
 - (2) Si la cláusula WHERE está ausente, el efecto es borrar TODAS las filas de la tabla.
 - (3) Como salida, al terminar exitosamente un DELETE se muestra en pantalla lo siguiente:
 - (a) DELETE (cont), donde *cont* es la cantidad de filas eliminadas.
- iv) SELECT * | columnas [...] FROM tabla WHERE condición ORDER BY [expresión ASC | DESC [...]]
 - (1) Esto devuelve la información solicitada de una tabla.
 - (2) Todos los elementos de la lista FROM con computados. Si más de un elemento es especificado en esta lista se hace un producto cartesiano de los datos.
 - (3) Si se especifica la cláusula WHERE todas las filas que no satisfagan la condición son eliminadas de la salida.
 - (a) La condición es una expresión que evalúa a un resultado de tipo booleano.
 - (4) Si la cláusula ORDER BY es especificada, las filas a devolver son ordenadas en el orden indicado. Si no se especifica un ORDER BY las filas se devuelven en cualquier orden que el sistema encuentre más fácil de producir.
 - (a) Donde la expresión puede ser el nombre de una columna especificada en el SELECT.
 - (b) Se puede agregar la palabra ASC para ascendente o DESC para descendente después de cualquier expresión en la cláusula ORDER BY. Si no se especifica se asume ASC de manera predeterminada.
 - (c) Los valores NULL se ordenan más alto que cualquier otro valor. Con un orden ascendente los valores NULL van al final y con un orden descendente los valores NULL van al principio.
 - (5) Como salida el SELECT debe mostrar los nombres de las columnas seleccionadas y los valores en el orden especificado. Al final también se debe indicar el número de filas que se muestran. Como referencia se pueden consultar las salidas del SELECT en PostgreSQL o las presentaciones de la clase.

TRABAJO A REALIZAR Y RECOMENDACIONES

- El lenguaje de programación queda a discreción de estudiante.
- No se calificarán los documentos de análisis y diseño pero es importante que se realicen previo al desarrollo
- El programa se debe realizar tomando en cuenta la eficiencia en el uso de memoria.
 - o Se deben poder utilizar archivos de prueba de más de 10,000 registros.
 - o Se debe cuidar el tiempo que tomar: insertar, borrar, actualizar o extraer más de 5,000 registros.
- Es muy importante la correcta presentación de los resultados.
- Se deben mostrar mensajes de instrucciones claros y mensajes de errores específicos y comprensibles.

III. Temas que refuerza

- SQL
- Instrucciones DDL
- Restricciones
- Instrucciones DML
- Insertar y actualizar
- SELECT y obtener datos
- Modelo físico de base de datos

IV. Material a enviar

- Código fuente en archivo .ZIP o .RAR
 - o El programa fuente y cualquier otro archivo indispensable para la correcta compilación y ejecución del programa. El proyecto solo será calificado si se puede ejecutar con normalidad.
 - o Programación clara y ordenada
 - o Incluir documentación interna:
 - Encabezado de cada archivo
 - Nombres de identificadores
 - Descripción de variables, constantes y métodos
 - Comentarios entre líneas
 - Tabulación y orden
 - o Readme.txt: documento que incluya aspectos especiales a tomar en consideración para la ejecución del programa.

V. Puntos extra

- Cualquier aspecto adicional a lo pedido en el proyecto, que sea de utilidad en el programa y tenga dificultad de programación, se tomará en consideración para puntos extra teniendo como máximo 10 puntos extra.

VI. Evaluación

- | | |
|--|--------------------------------|
| - Requerimientos entrega y documentación | 10pts |
| - DDL | 45pts (por pruebas sobre DBMS) |
| - DML | 45pts (por pruebas sobre DBMS) |
| TOTAL | 100pts |