

Administración de Bases de Datos

M.C. Carlos Rojas Sánchez¹

¹Licenciatura en Informática
Universidad del Mar - Puerto Escodido

[Bases de Datos II]

Objetivo

Proporcionar y aplicar los conocimientos fundamentales en la administración de bases de datos a través del diseño e implementación de esquemas utilizando el diseño relacional y orientado a objetos, garantizando la integridad, seguridad y recuperación de los datos.

1 Funciones y objetivos de la administración de bases de datos

Personajes

- Usuarios de los datos/aplicaciones
- Directivos de la propia organización
- Analistas y programadores
- Operadores
- Suministradores de SW y HW
- Personal de formación

Administración de BD's

- Diseño lógico y físico de bases de datos
- Asistir a las negociaciones para adquirir el SW y HW para el sistema de bases de datos
- Puesta en marcha del sistema nuevo en paralelo con el antiguo
- Monitorizar el funcionamiento de la BD: auditorías de seguridad, rendimiento, ocupación de espacios físicos, etc.
- Reorganizaciones físicas y lógicas de la BD
- Proponer y controlar mejoras de la BD

Administración de BD's

- Trabajar con el usuario
- Controlar aspectos relativos a la seguridad y confidencialidad
- Aspectos relativos a la integridad de los datos
- Definir procedimientos de recuperación y respaldo de la BD
- Conocer nuevos productos relacionados con los SBD
- Documentar los procedimientos de operación del SBD

2 Desarrollo, implantación y gerencia de una base de datos

Proceso de diseño

- Análisis y definición de requisitos
- Diseño conceptual de la base de datos
- Elección de un SGBD
- Paso a un modelo de datos o diseño lógico
- Diseño físico de la base de datos
- Implementación de la BD
- Carga de la base de datos

Tarea 1

- Principales SGBD
 - Costo / Beneficio
 - Ventajas / Desventajas
- Conceptos
 - CRUD
 - MVC

- 3 El administrador de base de datos (DBA)
 - Implementación del esquema interno y conceptual
 - Creación de usuarios, asignación y revocación de permisos
 - Implementación de las restricciones de seguridad

Función Asegurar que la información precisa y consistente esté disponible para los diferentes usuarios y para las aplicaciones cuando la necesiten y en la forma requerida

- Mismos datos integrados en una BD
- Distintos Usuarios
- Diferentes Usos

DBA

Tipos de Funciones

- Mantener Seguridad y Confidencialidad en el acceso a los Datos
- Recuperación de la BD en caso de fallo (errores de usuario, de sentencias, fallo en procesos, red, instancias o sistema)
- Mantener la Integridad y Consistencia de los datos y permitir Acceso Concurrente a los datos
- Realizar el Diseño Físico de la BD para optimizar recursos (tiempo de acceso, espacio en memoria, etc.)

DBA

Personal que interactúa con él (1)

- con usuarios en general: para especificar los requisitos que debe cumplir la BD, estudiar las necesidades y la mejor manera de satisfacerlas.
- con directivos: que fijarán los objetivos de la BD para que ésta responda a los objetivos generales de la empresa.
- con analistas y programadores, a los que el ABD deberá proporcionar información acerca de la estructura de los datos, especificaciones para las aplicaciones que vayan a desarrollarse, etc.
- con operadores, con ellos, el ABD deberá analizar los procedimientos de recuperación a aplicar en la explotación de la BD ante fallos del sistema, normas para copias de seguridad, reorganización, etc.

DBA

Personal que interactua con él (2)

- con suministradores, para corregir problemas que puedan producirse en el sistema de base de datos, además de estar al corriente de nuevas herramientas, interfaces, programas, etc que estén desarrollándose y que sean de interés en el entorno de la BD.
- con personal de formación, a fin de preparar los planes adecuados de formación sobre bases de datos a todos los niveles de la empresa.

Implementación

- Tener un SGBD
- Se basa en el diagrama E-R
- La implementación de la base de datos se realiza mediante las sentencias del lenguaje de definición de datos (LDD) del SGBD escogido. Estas sentencias se encargan de crear el esquema de la base de datos, los ficheros en donde se almacenarán los datos y las vistas de los usuarios

Implementación

- Las transacciones sobre la base de datos, se implementan mediante el lenguaje de manejo de datos (LMD) del SGBD
- También se implementan los menús, los formularios para la introducción de datos y los informes de visualización de datos mediante lenguajes de consultas no procedurales, generadores de informes, generadores de formularios, generadores de aplicaciones. Finalmente se implementan todos los controles de seguridad e integridad



MYSQL

DATOS

- Desarrollado en con C y C++
- Usado por Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr, y YouTube
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice
- Tablas hash en memoria temporales
- MySQL Server tiene bases de datos de hasta 50 millones de registros

Acceso a MySQL

`mysql -h NombreDelServidor -u NombreDeUsuario -p`

- `mysql -h 127.0.0.1 -u root -p`

MySQL

Indicadores

Indicador	Significado
mysql>	Espera una nueva sentencia
->	La sentencia aún no se ha terminado con ;
">	Una cadena en comillas dobles no se ha cerrado
'>	Una cadena en comillas simples no se ha cerrado

MySQL

Expresiones y variables

- MySQL dispone de variables de sesión, visibles únicamente durante la co- nexión actual. Éstas pueden almacenar valores de tipos enteros, flotantes o cadenas, pero no tablas
`select @x := 1;`

- Las variables locales permiten almacenar datos entre consultas
`select @hora_i := now();`
`select now() - @hora_i;`

MySQL

Bases de Datos

- `show databases;`
Para ver las bases de datos existentes en el sistema
- `create database nombre;`
Crear una nueva base de datos
- `drop database nombre;`
Elimina una base de datos

MySQL

Bases de Datos

- `select database();`
Informa sobre la base de datos actualmente en uso
- `use base_de_datos;`
Permite abrir una base de datos para su uso
- `describe tabla;`
Permite consultar las columnas de una tabla

- Crear una tabla
create table personas (
-> nombre varchar(40) not null,
-> dirección varchar(50) null, }
-> edo_civil char(13) default 'Soltero',
-> num_registro int primary key auto_increment,
->);
- Eliminar una tabla
drop table personas;
drop table if exists personas;
- Cambiar de nombre la tabla
alter table personal rename gente;

- Agregar y eliminar columnas
alter table personal add (
-> mascota char(30) default 'perro',
-> pasatiempo char (20) not null
->);
alter table personal drop pasatiempo;
- Modificar columnas
 - Redefinir
alter table personal modify
-> mascota char (14) default 'gato';
 - Cambiar nombre de la columna
alter table personal change nom
-> nombre char(20);
 - Ordenación física
alter table personal order by nom;

Usuarios

MySQL

- No se pueden crear usuarios sin asignarle al mismo tiempo privilegios
- MySQL permite definir diferentes usuarios, y además, asignar a cada uno determinados privilegios en distintos niveles o categorías de ellos

Niveles de privilegios

MySQL

- Globales: se aplican al conjunto de todas las bases de datos en un servidor
- De base de datos: se refieren a bases de datos individuales
- De tabla: se aplican a tablas individuales, y por lo tanto, a todas las columnas de esas tabla
- De columna: se aplican a una columna en una tabla concreta
- De rutina: se aplican a los procedimientos almacenados

Creación de Usuarios

CREATE USER y GRANT

Es preferible usar GRANT, ya que si se crea un usuario mediante CREATE USER, posteriormente hay que usar una sentencia GRANT para concederle privilegios

GRANT podemos crear un usuario y al mismo tiempo concederle también los privilegios que tendrá

Creación de Usuarios

GRANT

- *GRANT priv_type [(column_list)] [, priv_type [(column_list)]] ...
ON
TO user [IDENTIFIED BY [PASSWORD] 'password']
[, user [IDENTIFIED BY [PASSWORD] 'password']] ...*

La primera parte *priv_type [(column_list)]* permite definir el tipo de privilegio concedido para determinadas columnas. La segunda *ON {tbl_name | * | *.* | db_name.*}*, permite conceder privilegios en niveles globales, de base de datos o de tablas.

Creación de Usuarios

GRANT

- Para crear un usuario sin privilegios usaremos la sentencia:
`GRANT USAGE ON *.* TO anonimo IDENTIFIED BY 'clave';`
- Mostrar los privilegios de un usuario
`SHOW GRANTS FOR anonimo;`

Quitar privilegios de Usuarios

REVOKE

```
revoke all  
on *.*  
from visitante@localhost;
```

Operaciones sobre Usuarios

- Eliminar usuarios
drop user visitante;
- Cambiar el nombre de un usuario
RENAME USER viejo_usuario TO nuevo_usuario [,
viejo_usuario TO nuevo_usuario] ...
Ejemplo: RENAME USER Pepito TO Pepito@127.0.0.1;
- Cambiar la contraseña de un usuario
SET PASSWORD FOR usuario =
PASSWORD('contraseña');
Ejemplo: SET PASSWORD FOR Pepito =
PASSWORD('Grillo');

Establecer conexiones seguras

MySQL puede establecer conexiones seguras encriptándolas mediante el protocolo SSL*; de esta manera, los datos que se transmiten (tanto la consulta, en un sentido, como el resultado, en el otro) entre el cliente y el servidor estarán protegidos contra intrusos. Para especificar que un usuario debe conectarse:

```
grant all  
-> on *.*  
-> to visitante@localhost  
-> require ssl;
```

Limites de uso

Los recursos físicos del servidor siempre son limitados: si se conectan muchos usuarios al mismo tiempo al servidor y realizan consultas o manipulaciones de datos complejas, es probable que pueda decaer el rendimiento notablemente. Una posible solución a este problema es limitar a los usuarios el trabajo que pueden pedir al servidor con tres parámetros:

- Máximo número de conexiones por hora
- Máximo número de consultas por hora
- Máximo número de actualizaciones por hora

Limites de uso

grant all

-> on *.*

-> to

-> with MAX_CONECTIONS_PER_HOUR 3

-> MAX_QUERIES_PER_HOUR 300

-> MAX_UPDATES_PER_HOUR 30;

Respaldos

Opción 1

Un mecanismo que permite realizar la copia de los archivos del servidor sin necesidad de detener el servicio es el script 'mysqlhotcopy'. El script está escrito en Perl y bloquea las tablas mientras realiza el respaldo para evitar su modificación.

- `mysqlhotcopy demo /algun_directorio`

Respaldos

Opción 2

El comando `mysqldump`, realiza un volcado de las bases de datos pero traduciéndolas a SQL; es decir, entrega un archivo de texto con todos los comandos necesarios para volver a reconstruir las bases de datos, sus tablas y sus datos.

- `mysqldump demo > demo.sql`

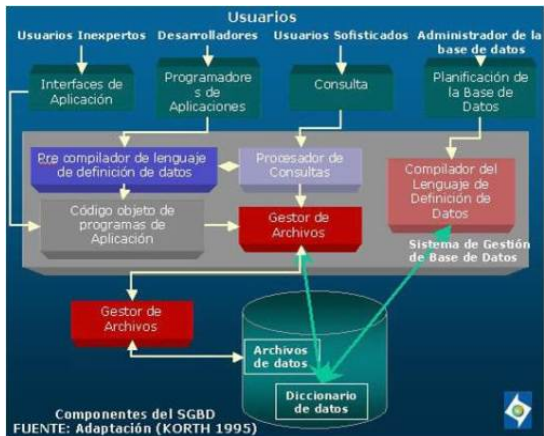
4 Gestores de bases de datos

Uso y evaluación de gestores de bases de datos

Optimización y ejecución

Portabilidad y migración de datos

SGBD



SGBD

- Procesador de Consultas: traduce sentencias de un lenguaje de consultas a otro de bajo nivel.
- Gestor de la Base de Datos: proporciona la interface entre los datos; almacenados en la base de datos y los programas de aplicación y las consultas que se realizan en el sistema.
- Gestor de Archivos: gestiona el espacio en memoria del disco usadas para almacenar la información.

SGBD

- Pre compilador del Lenguaje de Manipulación de Datos DML: convierte las sentencias DML de un programa de aplicación en llamadas normales a procedimientos.
- Compilador del Lenguaje de Definición de Datos DDL: convierte sentencias DDL en un conjunto de datos sobre datos.
- Gestor de Diccionario de Datos: Almacena metadatos sobre la estructura de la base de datos.

SGBD

- Control de Autorización: Comprueba que el usuario tiene los permisos necesarios para acceder a cierta información y realizar algunos cambios pertinentes. Ejemplo: contraseña y nombre de usuario.
- Procesador de Comandos: Una vez verificados los permisos al usuario se pasa el control al procesador de comandos.
- Control de la Integridad: cuando son modificados algunos datos, este módulo debe comprobar que la operación se realice de manera satisfactoria y cumpla con las restricciones de integridad.

SGBD

- Optimizador de Consultas: determina la estrategia mas adecuada para la realización de consultas.
- Gestor de Transacciones: este módulo realiza el procesamiento de transacciones.
- Planificador (scheduler): es el responsable de asegurar que las operaciones que se realizan concurrentemente sobre la base de datos tienen lugar sin conflictos.

SGBD

- Gestor de Recuperación: garantiza la recuperación de la información cuando se presente una falla.
- Gestor de Buffers: es el responsable de transferir los datos entre la memoria principal y los dispositivos de almacenamiento secundario.

Optimización

Indexación

La indexación es la principal herramienta para optimizar el rendimiento general de cualquier base de datos. Es también la más conocida por los usuarios de servidores MySQL y, paradójicamente, su no utilización es una de las principales causas de bajo rendimiento en servidores de bases de datos.

Optimización

Indexación

Ventajas:

- Permite ordenar las tablas por varios criterios simultáneamente.
- Es menos costoso ordenar un archivo índice, porque incluye sólo referencias a la información y no la información en sí.
- El costo de inserción y eliminación es menor.
- Con los registros siempre ordenados se utilizaran algoritmos mucho más eficientes que el simple recorrido secuencial en las consultas.

Optimización

Indexación

Desventajas:

- Los índices ocupan espacio en disco.
- Aún teniendo registros pequeños, el mantener en orden un índice disminuye la velocidad de las operaciones de escritura sobre la tabla.

Optimización

La cache de consultas de MySQL

El servidor MySQL incluye la posibilidad de utilizar una cache con los resultados de las últimas consultas para acelerar la velocidad de respuesta. Esta solución es útil cuando las tablas tienen relativamente pocos cambios y se realizan los mismos tipos de consultas.

Optimización

La cache de consultas de MySQL

Funcionamiento

- La primera vez que se recibe una consulta se almacena en la cache.
- Las siguientes veces la consulta se realiza primero en la cache; si tiene éxito, el resultado se envía inmediatamente.

Optimización

La cache de consultas de MySQL

Características

- El servidor compara el texto de la consulta; aunque técnicamente sea igual si difiere en uso de mayúsculas-minúsculas o cualquier otro cambio, no se considera la solicitud idéntica y no será tratada por la cache.
- Si alguna tabla incluida en alguna consulta cambia, el contenido de la consulta es eliminado de la cache.

Optimización

La cache de consultas de MySQL

Configuración

- `query_cache_limit`. No almacena resultados que sobrepasen dicho tamaño. Por omisión es de 1M.
- `query_cache_size`. Tamaño de la memoria cache expresada en bytes. Por omisión es 0; es decir, no hay cache.
- `query_cache_type`. Puede tener tres valores: ON , OFF o DEMAND.

Bibliografía I



Jorge Sánchez.

Principios sobre Bases de Datos Relacionales.

Creative-Commons, 2004.



Abraham Silberschatz, Henry F. Korth and S. Sudarshan.

Fundamentos de Bases de Datos.

McGRAW-HILL, 2002.