



Preparador Informática

www.preparadorinformatica.com

TEMA 43. INFORMÁTICA

ADMINISTRACIÓN DE SISTEMAS DE
BASE DE DATOS.

TEMA 43 INF: ADMINISTRACIÓN DE SISTEMAS DE BASE DE DATOS.

1. INTRODUCCIÓN
2. CONCEPTOS BÁSICOS
3. ADMINISTRACIÓN DE SISTEMAS DE BASE DE DATOS
 - 3.1. OBJETIVOS Y FUNCIONES DEL ADMINISTRADOR DE LA BASE DE DATOS
 - 3.2. HERRAMIENTAS DE ADMINISTRACIÓN
4. GESTIÓN DEL ALMACENAMIENTO
5. SEGURIDAD DE LA BASE DE DATOS
 - 5.1. SEGURIDAD EN SQL
6. TRANSACCIONES. RECUPERACIÓN Y CONCURRENCIA
 - 6.1. RECUPERACIÓN DEL SISTEMA
 - 6.2. PROBLEMAS DE CONCURRENCIA
7. CONCLUSIÓN
8. BIBLIOGRAFÍA



1. INTRODUCCIÓN

Hoy en día el mayor activo de las organizaciones son los datos y su gestión eficaz y segura. Por ello, si analizamos la mayoría de los ámbitos de actividad, nos encontramos que la utilización de las bases de datos está ampliamente extendida (al registrarse en una web, al acudir a la consulta médica, al consultar el catálogo de productos de una tienda online, etc.). Las bases de datos y los datos contenidos en ellas, son imprescindibles para llevar a cabo multitud de acciones. La necesidad de almacenar datos de forma masiva dio paso a la creación de los sistemas de bases de datos.

Para poder gestionar adecuadamente estos sistemas de bases de datos es imprescindible realizar una buena administración del sistema de base datos resaltando la figura del Administrador de la Base de Datos.

Lo expuesto anteriormente justifica la importancia del presente tema donde vamos a empezar definiendo una base de datos y un sistema gestor de base de datos para posteriormente centrarnos en los conceptos más importantes de la administración de los sistemas de bases de datos.

2. CONCEPTOS BÁSICOS

Podemos definir una **base de datos** como un conjunto, colección o depósito de datos almacenados en un soporte informático no volátil. Los datos están interrelacionados y estructurados.

Un **sistema gestor de base de datos o SGBD** es una colección de programas de aplicación que permite a los usuarios la creación y el mantenimiento de una base de datos, facilitando la definición, construcción y manipulación de la información contenida en ésta.

3. ADMINISTRACIÓN DE SISTEMAS DE BASES DE DATOS

La administración de la BD se ocupa básicamente de asegurar que la información esté disponible para los usuarios y aplicaciones en el instante



preciso y en la forma adecuada. Esta información ha de proporcionar dos características básicas: precisión y consistencia.

El Administrador de la Base de Datos (ABD ó DBA) interactúa tanto con el sistema como con los usuarios para llevar a cabo las tareas que permitan la puesta a punto y mantenimiento de las bases de datos.

3.1. OBJETIVOS Y FUNCIONES DEL ADMINISTRADOR DE LA BASE DE DATOS.

Los objetivos y funciones para un administrador de la base de datos relacionados entre ellos son:

Comunicación con los usuarios de la base de datos y responsables del sistema:

- Detectar necesidades de uso, roles, privilegios, etc
- Determinar la interfaz de conexión con otros sistemas ya existentes.
- Identificar recursos hardware y software disponibles en la organización y en su caso adoptarlos. Caso de no ser suficientes proponer su ampliación y/o mejora.
- Determinar la infraestructura de red existente y su posible ampliación en función de las nuevas demandas.

Planificación, diseño e implementación de los sistemas de bases de datos:

- Obtener los cuadernos de carga elaborados por los analistas y diseñadores donde se recogen los detalles de implantación de la BD.
- Desarrollar los scripts de creación de la BD.
- Desarrollar los scripts que permitan la puesta a punto de la BD.

Establecimiento de normas y procedimientos:

- Establecer normas y procedimientos para controlar la seguridad y la integridad de los datos de forma eficiente. Las normas se aplican particularmente al control del desarrollo y del uso de la programación y de las operaciones de la base de datos.

- En el área de la programación, las normas se establecen para asegurar que los programas se revisen y se prueben antes de ponerlos en producción.
- En el área de las operaciones, las normas pueden establecerse para mantener los diarios de las operaciones de las transacciones y se crean procedimientos para la corrección de los errores, para el tratamiento de los puntos de control y para garantizar la copia de seguridad y la recuperación.

Mantener la integridad de los datos:

- Protección ante accidentes tales como los errores en la entrada de los datos o en la programación, del uso malintencionado de la base de datos y de los fallos del hardware o del software.
- Garantizar la actualización de los datos bajo los principios de atomicidad derivados del uso de transacciones.

Mantener la seguridad de los datos:

- Garantizar el acceso a aquellos usuario y/o aplicaciones que estén reconocidos e identificados por el sistema.
- Determinar el nivel de seguridad exigible.

Mantener la disponibilidad de los datos:

- Establecer los componentes que se han de mantener en sistemas redundantes con el fin de garantizar su uso continuado.
- Procedimientos de recuperación ante posibles pérdidas de información.
- Determinar la necesidad de gestionar el almacenamiento mediante roles independientes.
- Arranque/parada de los procesos/servicios asociados a la BD.

Carga de la BBDD:

- Desarrollar y ejecutar los scripts que permitan llevar a cabo la migración de los datos normalmente existentes en algún otro sistema de almacenamiento de información (ficheros, otros SGBDs, etc).

Mantenimiento de la BBDD:

- Adecuar los parámetros de la base a las exigencias cambiantes de cada momento, dispositivos de almacenamiento, área de memoria... con el fin de garantizar la eficiencia en el acceso a la información.
- Creación de nuevos usuarios, grupos, roles...
- Modificación de objetos ya existentes, o en su caso creación de nuevos.
- Optimización del diccionario de datos y consultas.
- Políticas de salvaguarda y recuperación periódicas.

3.2. HERRAMIENTAS DE ADMINISTRACIÓN.

El administrador de la base de datos puede utilizar distintas herramientas:

- Lenguajes de definición de datos: de manera que pueda especificar todos los objetos, atributos e interrelaciones que se almacenarán en la BD, su organización física en los soportes de ordenador, vistas de usuario y las restricciones de integridad y de confidencialidad.
- Utilidades del SGBD: para copias de seguridad, carga y descarga de datos de la base a ficheros externos y viceversa.
- Simuladores y monitores: que permitirán estudiar distintas alternativas de instrumentación física con el fin de optimizar la BD.
- Herramientas de ayuda al diseño: como las herramientas CASE que dan soporte para el proceso de diseño de la BD.
- Diccionarios de datos: con información relativa a los datos manejados por la organización.

4. GESTIÓN DEL ALMACENAMIENTO

Toda base de datos se graba en algún sistema de almacenamiento secundario constituido por uno o varios ficheros físicos soportados por el sistema de ficheros presente en la arquitectura hardware de soporte al SGBD. Cada fichero únicamente puede estar asociado a una base de datos. La gestión del espacio en disco es una de las grandes preocupaciones de cualquier administrador de base de datos. Según la forma de realojar más espacio para los objetos tendrá



una incidencia directa en su crecimiento, afectando al rendimiento de la base de datos.

En relación con la gestión del almacenamiento, el administrador de la base de datos perseguirá los siguientes objetivos teniendo en cuenta los requisitos de los procesos, las características del SGBD y las características del sistema operativo y del hardware:

- Disminuir los tiempos de respuesta.
- Minimizar el espacio de almacenamiento.
- Evitar las reorganizaciones costosas en tiempo y en recursos.
- Proporcionar el máximo de seguridad.
- Optimizar el consumo de recursos.

5. SEGURIDAD DE LA BASE DE DATOS

Los mecanismos de seguridad deben garantizar que los usuarios solo pueden realizar aquellas operaciones para las que están autorizados sobre ciertos objetos particulares. Por otro lado, hay que tener en cuenta que distintos usuarios pueden tener diferentes tipos de autorizaciones sobre los mismos objetos.

Las medidas de seguridad de la información las toma el administrador de la base de datos, para lo que utiliza las herramientas proporcionadas por el lenguaje al efecto.

5.1. SEGURIDAD EN SQL

En el caso de SQL, el sistema cuenta con diferentes mecanismos implicados en el mantenimiento de la seguridad:

- El sistema de gestión de vistas.
- El subsistema de autorización, mediante el cual usuarios con derechos específicos pueden conceder de manera selectiva y dinámica esos derechos a otros usuarios, y su revocación.
- Control de transacciones.



Las vistas pueden ser utilizadas con propósitos de seguridad, delimitando el acceso de los usuarios a ciertas porciones de la información.

```
CREATE VIEW [user.]view [ ( alias [, alias] ... ) ]
AS query
[WITH CHECK OPTION [CONSTRAINT constraint] ]
```

Es importante destacar que el uso de "WITH CHECK OPTION" garantiza que se puedan insertar registros en la tabla, a través de la vista, sólo si se cumple la condición de restricción especificada.

Para completar los mecanismos de seguridad es necesario disponer de ciertas órdenes que nos permitan asignar a los usuarios permisos de acceso a tablas y vistas de la base de datos. Para ello se utilizan las sentencias GRANT y REVOKE cuya sintaxis es la siguiente:

GRANT:

GRANT privilegios

ON elemento

TO nombreUsuario IDENTIFIED BY 'contraseña'

[WITH GRANT OPTION];

Donde:

- *privilegios puede ser ALL PRIVILEGES, para asignar todos los privilegios excepto GRANT, o una lista de ordenes SQL separadas por coma (SELECT,INSERT,DELETE,UPDATE).*
- *Elemento puede ser:*
 - ** para señalar todas las bases de datos.*
 - *baseDeDatos * todas las tablas de la base de datos.*
 - *baseDeDatos.Tabla una tabla determinada de una base de datos concreta*
- *En cuanto a al nombre de usuario, siempre se escribe nombreUsuario@nombrehost.*
- *IDENTIFIED, sirve para definir la contraseña del usuario.*
- *WITH GRANT OPTION, es opcional y permite asignar al usuario privilegios para crear otros usuarios.*

EJEMPLOS:

- *GRANT ALL PRIVILEGES ON * to administrador@localhost IDENTIFIED BY "JOSE";*

- *GRANT SELECT, INSERT, UPDATE ON facturacion.* to web@localhost IDENTIFIED BY "jose";*
- *GRANT SELECT, INSERT, UPDATE ON facturacion.clientes to comercial@localhost IDENTIFIED BY "Mercedes";*

REVOKE:

REVOKE privilegios

ON elemento

FROM nombre_de_usuario;

Donde:

- *Los privilegios son los mismos que en el caso de GRANT, más el GRANT OPTION.*

EJEMPLO:

- *REVOKE ALL PRIVILEGES ON * FROM administrador@localhost*

6. TRANSACCIONES. RECUPERACIÓN Y CONCURRENCIA.

Una transacción es una unidad lógica de trabajo. Es una secuencia de operaciones en una base de datos mediante la cual un estado consistente de la base de datos se transforma en otro estado consistente, sin conservar por fuerza la consistencia en todos los estados intermedios.

Para ello tenemos un componente del sistema encargado de este propósito denominado "gestor de transacciones", y las operaciones en SQL son COMMIT y ROLLBACK.

La mayoría de los SGBD incluyen las funciones rollback, commit y autocommit. Supongamos que queremos borrar una fila de una tabla, pero, al teclear la orden SQL, se nos olvida la cláusula WHERE y borramos todas las filas de la tabla. Esto no es problema pues podemos dar marcha atrás a un trabajo realizado mediante la orden ROLLBACK, siempre y cuando no hayamos validado los cambios en la base de datos mediante la orden COMMIT.

Cuando hacemos transacciones sobre la base de datos, es decir, cuando insertamos, actualizamos y eliminamos datos en las tablas, los cambios no se aplicarán a la base de datos hasta que no hagamos un COMMIT. Esto significa



que, si durante el tiempo que hemos estado realizando transacciones, no hemos hecho ningún COMMIT y de pronto se va la luz, todo el trabajo se habrá perdido, y nuestras tablas estarán en la situación de partida.

Para validar los cambios que se hagan en la base de datos tenemos que ejecutar la orden COMMIT:

COMMIT;

En algunos sistemas se nos permite validar automáticamente las transacciones sin tener que indicarlo de forma explícita. Para eso sirve el parámetro AUTOCOMMIT. El valor de este parámetro se puede mostrar con la orden SHOW, de la siguiente manera:

SHOW AUTOCOMMIT;

OFF es el valor por omisión, de manera que las transacciones (INSERT, UPDATE y DELETE) no son definitivas hasta que no hagamos COMMIT. Si queremos que INSERT, UPDATE Y DELETE tengan un carácter definitivo sin necesidad de realizar la validación COMMIT, hemos de activar el parámetro AUTOCOMMIT con la orden SET:

SET AUTOCOMMIT ON;

Ahora, cualquier INSERT, UPDATE y DELETE se validará automáticamente.

La orden ROLLBACK aborta la transacción volviendo a la situación de las tablas de la base de datos desde el último COMMIT siempre y cuando no esté activado el AUTOCOMMIT:

ROLLBACK;

Para asegurar la integridad de los datos se necesita que el sistema de base de datos mantenga las siguientes propiedades en las transacciones:

- Atomicidad.
- Consistencia.
- Aislamiento.
- Durabilidad.

6.1. RECUPERACIÓN DEL SISTEMA.

El sistema debe estar preparado para recuperarse no solo de fallos puramente locales, sino también de fallos globales. Un fallo local afecta sólo a la transacción en la que se presentó el fallo, mientras que un fallo global afecta a todas las transacciones que se estaban realizando en el momento del fallo.

Ante un fallo global, el sistema deberá recuperarse de dos tipos de fallos:

- Fallos del sistema, que afectan a todas las transacciones que se están realizando, pero no dañan físicamente a la base de datos. También se conocen como caídas suaves.
- Fallos de los medios de almacenamiento, que causan daños a la base de datos o a una porción de ella, y afectan al menos a las transacciones que están utilizando esa porción. También se conocen como caídas duras.

6.2. PROBLEMAS DE CONCURRENCIA

La mayoría de los SGBD son multiusuario. En estos sistemas se necesita un mecanismo de control de concurrencia para asegurar que ninguna transacción concurrente interfiera con las operaciones de las demás.

Son tres las situaciones en las que una transacción, aunque correcta en sí, puede producir de todos modos un resultado incorrecto debido a una interferencia por parte de alguna otra transacción:

- El problema de la modificación perdida.
- El problema de la dependencia no comprometida.
- El problema del análisis inconsistente.

El bloqueo es la técnica más usual que se utiliza para resolver problemas de concurrencia. La noción básica de bloqueo es simple: cuando una transacción requiere la seguridad de que algún objeto en la cual está interesada no cambiará de alguna manera no predecible sin que ella se dé cuenta, adquiere un bloqueo sobre ese objeto, con lo que ninguna transacción podrá leer ni modificar dicho objeto.

7. CONCLUSIÓN

En este tema hemos podido ver como toda la información que encontramos en una base de datos está sujeta a varios peligros, ya sean accidentales o provocados. Por esto el Sistema Gestor de Base de Datos debe tener controles para proteger la Base de Datos contra estos tipos de riesgos. El SGBD debe vigilar las peticiones del usuario y rechazar todo intento de violar las restricciones de seguridad y de integridad. Para ello es imprescindible realizar una buena administración de los sistemas de bases de datos y es necesario que el administrador de la base de datos realice todas sus funciones correctamente y cumpla con los objetivos requeridos.

8. BIBLIOGRAFÍA

- Date D.J.: **Introducción a los sistemas de bases de datos**. Editorial Addison-Wesley
- De Miguel A,y Piattini M:**Fundamentos y modelos de BBDD**. Edit. Ra-Ma
- Korth H. y Silberschatz: **Fundamentos de bases de datos**. Editorial McGraw-Hill
- http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro21/1_importancia_de_la_administracin_de_datos.html