

Unidad 1

Antonio Emiko Ochoa Adame

15 de febrero de 2019

Índice

1. Administración de una DB	1
1.1. Control de procesos repetitivos	1
1.2. Administración de derechos y restricciones	2
1.3. Procesamiento y memoria	2
1.4. Desarrollo de la seguridad de la base de datos	2
1.5. Acceso a la base de datos y seguridad a nivel de aplicación	2
1.6. Administración del SGDB (DBMS)	2
1.7. Mantenimiento del repositorio	3
1.8. Monitoreo	3
1.9. Mantenimiento	3
1.10. (11) Auditorías de logs	4
1.11. (12) Respaldo y restauración	4
1.12. (13) Concurrencia	4
1.13. (14) Bloqueos	5
1.14. (15) Problemas de pérdida de información	5
1.15. (16) Migraciones	5
2. RespalDOS	5

1. Administración de una DB

- Trooper habla más con el usuario.
- Skywalker soluciona problemas importantes (migraciones de datos, problemas más fuertes con la bases de datos).
- Vader sabe tratar a la gente, problemas con ingeniería social.

Las responsabilidades de DBA es facilitar el desarrollo y el uso de la base de datos.

- Diseñar el modelo lógico y físico.
- Configuración del servidor
- Etc.

1.1. Control de procesos repetitivos

Automatizar tareas, por ejemplo, backups.

Recomendación: Instalar **SQLServer 2017** (versión developer tiene más funciones).

1.2. Administración de derechos y restricciones

Qué se debe de hacer:

- Ir a recursos humanos.
- Matriz de permisos.

1.- Se hace un análisis 2.- Se genera una tabla llamada “matriz de permisos”, que sería hacer un análisis de los permisos, usuarios, cantidad de tablas en las bases de datos, etc.

1.3. Procesamiento y memoria

Se pueden asignar al DBMS la cantidad de recursos que puede utilizar (RAM y ROM). Se puede elegir la cantidad de procesadores para la DB para obtener un rendimiento adecuado en la DB.

NOTA

- Motor
- Servicio
- Instancia

Motor: Sería aqué que hace posible el servicio. Ayuda a hacer acciones en la DB.

Servicio: Permite abrir ventana. Intangible. Permite o inicia al DBMS y permite que el motor funcione con la instancia (gasolina de coche).

Instancia: “Área de trabajo donde se tienen las bases de datos; espacio lógico”. Organización de la base de datos.

Motor e instancia van ligados (moto-instancia).

Se pueden generar cualquier cantidad de instancias en cualquier DBMS.

En MySQL se puede usar “sysbench” para observar cosas relacionadas con los recursos.

1.4. Desarrollo de la seguridad de la base de datos

1.5. Acceso a la base de datos y seguridad a nivel de aplicación

Se tiene que ser conciente de cómo se lleva a cabo esta parte, incluso si no es responsabilidad directa del DBA. Se tiene que saber cómo funciona la seguridad desde el lado de la aplicación

1.6. Administración del SGDB (DBMS)

- Pago del licenciamiento
- Documento de proceso de instalación, desinstalación y actualización.
- Verificación de parches y actualizaciones.
- Documentación de ruta de instalación, servidores, direcciones IP, etc.
- Permisos de acceso a servidores de producción y pruebas.

1.7. Mantenimiento del repositorio

- Definición inicial de rutas de archivos primarios, secundarios, tablespaces, logs, unidades de disco.
- Monitoreo constante del espacio y crecimiento y definir un porcentaje cuando se deba levantar un alerta.
- Estadísticas mensuales.
- Documentar procesos de attach y detach y de migración.

Se hace Shrinking

Se cuentan objetos en la bases de datos (tablas, campos, tipo de datos, usuarios, etc.). Observar el crecimiento de la DB, obtener gráfica de crecimiento. Calcular la cantidad de tiempo que podría tardar en llenarse el almacenamiento.

Si la DB está en uso, se debe terminar de usar en la DB (voluntariamente o “a la fuerza”)

dbcc ayuda a darle mantenimiento a la DB.

1.8. Monitoreo

- Seguridad
- Queries en proceso
- Indexación

Seguridad: Agregar seguridad para que no hayan problemas con usuarios dentro de la misma organización o accesos no autorizados.

Queries en proceso: Se puede detectar si alguien está haciendo queries indebidos o queries no optimizados o “malos”.

Indexación:

1.9. Mantenimiento

- SHRINK LOG
- DBCC SHRINKDATABASE;
- DBCC SHRINKDATABASE (DB, TRUNCATEONLY);
- DBCC CHECKTABLE

Integridad:

La manera en la que está organizada la DB físicamente (en el disco).

Bloque 8 KB (un porción de una tabla).

Se necesitan varios **bloques** para guardar varias **tablas**.

Un conjunto de **bloques** se llama **Extensión**.

Varias **extensiones** forman un **segmento**.

DBCC CHECKTABLE ayuda a checar la integridad.

Indices:

Son archivos.

Se deben reorganizar para optimizar o que funcionen correctamente.

Limpieza de logs binarios MySQL.

Optimización de tablas MySQL (OPTIMIZE TABLE).

Vaciado de caché de consultas.(FLUSH QUERY CACHE).

Rotación de logs binarios MySQL. (Se vacían manualmente si ya no hay espacio). (Puede que tenga que ser creada una nueva bitácora)

1.10. (11) Auditorías de logs

Se pueden seleccionar que campos se desean auditar.

1.11. (12) Respaldo y restauración

RPO: Cuando se dan señales de que algo anda mal.

Desastre: (En el mejor escenario hay un respaldo).

RTO: Cuando se concluye que todo volverá a funcionar normalmente.

- Frecuencia
- Horario
- Rotación
- Bitácora
- Tareas automatizadas
- Caja fuerte
- Sitio alternativo

1.12. (13) Concurrencia

- Manejo de transacciones
- Llaves primarias
- Índices
- Propiedades ACID:
 - Atomic
 - Consistent
 - Isolated
 - Durable

Manejo de transacciones

Cuando hay un select se genera un “bloqueo compartido”. Un select se atiende más rápido porque solo es una lectura (bloqueo compartido).

Cuando hay un update se genera un “bloqueo exclusivo”.

Begin transaction, rollback y commit ayudan para asegurar la consistencia de la información.

Llaves primarias

Evita redundancia de tuplas

Índices

Ayuda a que los queries se ejecuten más rápido.

ACID

Deben asegurar que cuando se haga una transacción se debe asegurar ACID.

Definir ACID.

Durabilidad.- Si es exitosa la transacción, debe hacerse un commit para que los cambios permanezcan o “persistan”.

1.13. (14)Bloqueos

Exclusivo, compartido, deadlock.

Se bloquea cuando dos usuarios quieren utilizar el mismo recurso al mismo tiempo. A esto se le llama **deadlock**.

Se debe monitorear para ver si ocurren deadlocks.

Se debe liberar el deadlock manualmente, pero esto puede generar inconsistencia en la información.

1.14. (15)Problemas de pérdida de información

- Correcciones manuales automatizadas.
- Correcciones masivas.

1.15. (16)Migraciones

- De tabla a tabla.
- De una DB a otra DB.
- A una DB de producción.
- A una DB de consultas.
- A una DB de pruebas.
- DB distribuída.
- Implicaciones en llaves primarias y autoincrementables.

Cuando un campos se usa mucho se le puede generar un índice para optimizar una búsqueda.

Si no es suficiente, se puede partir la tabla para en campos importantes para poder agilizar los queries.

Una tabla pequeña puede crecer demasiado sin darnos cuentas, por es es importante sacar estadísticas, así se puede observar el comportamiento de la DB y poder tomar decisiones en base a dicha información.

2. Respaldos

Técnicas de respaldo.

Inicio de código

USE MASTER;

BACKUP DB-NAME

Fin de código

Página o bloque.- Unidad mínima de almacenamiento de una DB.

Varios bloques/páginas forman un extent se guardan tablas.

Varios extent forma un segmento.

Una DB siempre está dividido en 2 archivos: DATA y LOG.

Automatización de los respaldos es lo más adecuado.

Faltan temas: bak en MySQL.

Práctica: Se generará código a partir de un modelo relacional, generar, respaldos, procedures, trigger, tareas de mantenimiento y monitoreo, insertar tuplas, etc.