comandos

SYBASE

1. Comandos SYBASE

1.1 ASE

Crear un usuario en Solaris para administrar Sybase: **sybase**

Arrancar el servidor de BD:

$SYBASE/install/startserver –f $SYBASE/install/RUN_nombreMaq

Arrancar el servidor de Backups:

$SYBASE/install/startserver –f SYBASE/install/RUN_nombreMaq_back

Entrar en el isql como system administrator: **isql –SnombreMaquina –Usa**

1.2 Dispositivos

1. Comprobar el estado de los dispositivos actuales y los vdevno que hay libres.

- Mediante un procedimiento del sistema: **sp_helpdevice**
- Mediante queries sobre tablas del sistema.
  - **select * from sysdevices**
  - **select * from sysusages**
- Mediante una query especial que muestra los vdevno que están ocupados.
  - **select vdevno = (low/16777213),**
    nombre = name,

    tamaño = ((high-low) + 1/512),

    paginas = (high –low) + 1

    from sysdevices

    where cntrltype = 0

2. Crear dos dispositivos, uno para datos de 8MB de y otro para log de 2MB.

  - **disk init**
    name = “test_data_device”,

    physname = “/sybase/tests/test_data_device.dat”,

    vdevno = 5,

size = 4096

- **disk init**
  name = "test_log_device",

  physname = "/sybase/tests/test_log_device.log",

  vdevno = 6,

  size = 1024

3. Hacer que el dispositivo test_data_device sea el dispositivo por defecto.

   - **sp_diskdefault master, defaultoff**
   - **sp_diskdefault test_data_device, defaulton**

4. Comprobar el estado de los dispositivos.
   - **sp_helpdevices**

1.3 Mirror

1. Crear un mirror del dispositivo test_log_device en test_log_device.mirror

   - **disk mirror**
     name = "test_log_device",

     mirror = "/sybase/tests/test_log_device.mirror"

2. Parar el mirror

   - **disk unmirror**
     name = "test_log_device"

3. Rearrancar el mirror.

   - **disk remirror**
     name = "test_log_device"

1.4 Bases de Datos

1. Crear una base de datos de 10MB con 8MB de datos y 2 MB de log.

   - **create database database_test**
     on test_data_device = 8

     log on test_log_device = 2

2. Comprobar que la base de datos se ha creado correctamente.

   - **sp_helpdb database_test**

3. Cambiar la opción dbo_use_only a true en la base de datos creada.

   - **sp_dboption database_test, "dbo_use_only", true**

4. Escribir una query para mostrar las BD registradas en un dispositivo.

   - **select distinct**
     "physical device" = convert (char(40), sysdevice.phyname),

     "database name" = convert (char(15), sysdatabases.name)

   from sysdevices, sysdatabases, sysusages

   where sysdevices.name = "X"

     and sysdevices.dbid = sysdatabases.dbid

     and sysusages.vstart

   between sysdevices.low and sysdevices.high

1. Borrar todo lo que hemos hecho hasta ahora.

   - **use master**
   - **drop database database_test**
   - **sp_dropdevice test_data_device**
   - **sp_dropdevice test_log_device**

   Borrar los ficheros de los dispositivos creados en /sybase/tests

   Apagar el servidor y volver a arrancarlo.

2. Crear 3 dispositivos. Uno de datos de 10MB, otro de log de 4MB y otro de índices de 2MB.

   - **disk init**
     name = "test_data_device",

     physname = "/sybase/tests/test_data_device.dat",

     vdevno = 5,

     size = 5120

   - **disk init**
     name = "test_log_device",

     physname = "/sybase/tests/test_log_device.log",

     vdevno = 6,

     size = 2048

   - **disk init**
     name = "test_index_device",

     physname = "/sybase/tests/test_index_device.log",

     vdevno = 7,

     size = 1024

3. Crear la base de datos con 8MB de datos y 2MB de log.

   - **create database database_test**
     on test_data_device = 8

     log on test_log_device = 2

4. Ver los segmentos de la base de datos.

   - **sp_helpdb database_test**

5. Ampliar el dispositivo de datos de la BD a 2MB más para completar todo el espacio.

   - **alter database database_test on test_data_device = 2**

6. Ampliar la BD para que utilice también el dispositivo de indices.

   - **alter database database_test on test_index_device = 1**

   Al hacer esto se crean los segmentos system y default en el dispositivo.

7. Crear el segmento seg_test en el dispositivo de índices.

   - **use database_test**
   - **sp_addsegment seg_test, database_test, test_index_device**
   - **sp_helpsegment**

8. Borrar los segmentos del sistema en el index (no se necesitan).

   - **sp_dropsegment "system", database_test, test_index_device**
   - **sp_dropsegment "default", database_test, test_index_device**

9. Crear una tabla de ejemplo

   - **create table ejemplo (campo1 int, campo2 int)**

- **sp_help ejemplo**

10. Crear un índice para el campo1 de la tabla.

- **create index ejemplo_index on ejemplo(campo1) on seg_test**

11. Comprobar que se ha creado todo correctamente.

- **sp_helpsegment "default" : aquí está la tabla ejemplo**
- **sp_helpsegment "seg_test" : aquí está el índice ejemplo_index**

12. Crear un índice que contiene los datos

- **create clustered index cluster_index**
  on ejemplo (campo1)

  on seg_test

13 Crear una query para obtener los fragmentos usados por un determinado segmento.

- **select segmento=seg.name, fragmentos=count(*), tamaño=sum(size)/512**
  from master..sysusages u, syssegments seg

  where segmap <= 7 and db_id()=dbid

  group by seg.name

1.6 Particiones

1. Crear 4 dispositivos

   - **disk init**
     name = "dd1",

     physname = "/sybase/tests/dd1.dat",

     vdevno = 11,

     size = 1024

   - **disk init**
     name = "ld1",

     physname = "/sybase/tests/ld1.log",

     vdevno = 12,

     size = 512

   - **disk init**
     name = "dd2",

     physname = "/sybase/tests/dd2.dat",

     vdevno = 13,

     size = 1024

   - **disk init**
     name = "dd3",

     physname = "/sybase/tests/dd3.dat",

     vdevno = 14,

     size = 1024

2. Crear una BD

- **create database partition_test**
  on dd1 = 2

  log on ld1 = 1

- **alter database partition_test**
  on dd2 = 2, dd3 = 2

3. Definir los segmentos

- **use partition_test**
- **sp_addsegment partition_seg, partition_test, dd2**
- **sp_extendsegment partition_seg, partition_test, dd3**
- **sp_dropsegment "default", partition_test, dd2**
- **sp_dropsegment "default", partition_test, dd3**
- **sp_dropsegment "system", partition_test, dd2**
- **sp_dropsegment "system", partition_test, dd3**
- **sp_helpsegment partition_seg**

4. Crear una tabla particionada en dd3

- Crear tabla

- **create table tabla_ejemplo (campo1 int, campo2 char(10)) on partition_seg**
- **sp_help tabla_ejemplo**

- Particionar la tabla

- **alter table tabla_ejemplo partition 3**
- **sp_help tabla_ejemplo**

- Páginas en cada partición

- **select partitionid,**
  ptn_data_pgs(object_id("tabla_ejemplo"), partitionid) Pags

  from syspartitions

  where id = object_id("tabla_ejemplo")

1.7 Usuarios

1. Crear 2 logins para poder acceder a database_test

   - **use master**
   - **sp_addlogin login1, password1, database_test**
   - **sp_addlogin login2, password2, database_test**

2. Crear dos usuarios que se correspondan con los dos logins anteriores

   - **use database_test**
   - **sp_adduser login1, user1**
   - **sp_adduser login2, user2**

3. Crear un grupo de usuarios en la BD

   - **sp_addgroup grupo_test**

4. Añadir los dos usuarios creados al grupo

   - **sp_changegroup grupo_test, user1**
   - **sp_changegroup grupo_test, user2**

5. Comprobar que la creación del grupo y los usuarios ha sido correcta.

   - **sp_helpgroup**
   - **sp_helpgroup grupo_test**

6. Cambiar la password al login1

   - **sp_password password_antigua, password_nueva, login1**

1.8 Conceder y Revocar permisos

1. Crear una tabla de test

   - **use database_test**
   - **create table gr_tabla(i int, c char(24))**

2. Ejecutar proc filltable para ver los datos

```
rm -f "6d:filltable.sql"

val=10

while [ "$val" -le 30 ]

do
```

   - **create proc gr_tabla_proc**
     as

     print "Las tuplas de la tabla gr_table son: "

     select * from gr_tabla

   - **exec gr_tabla_proc**

3. Añadir nuevos usuarios para las pruebas

   - **use database_test**
   - **sp_adduser login1, nuevo1**
   - **sp_adduser login2, nuevo2**

4. Ver los logins existentes.

   - **select name from master..syslogins**

5. Ver usuarios en esta db

   - **select name from sysusers**

- **sp_helpuser**

6. Añadir el alias para el login prueba

   - **sp_addalias prueba, nuevo1**

7. Verificar con sp_helpuser

   - **select * from sysalternates**

8. Conceder select, insert y update sobre la tabla a nuevo1

   - **grant select, insert, update on gr_tabla to nuevo1**

9. Conceder execute sobre el procedimiento a nuevo2

   - **grant execute on gr_tabla_proc to nuevo2**

10. Conceder select sobre la primera columna de la tabla a user1

    - **grant select on gr_tabla(i) to nuevo2**
    - **sp_helprotect gr_tabla**

11. Crear un grupo grupo_test y añadir los usuarios nuevo1 y nuevo2.

    - **sp_addgroup grupo_test**
    - **sp_changegroup grupo_test, nuevo1**
    - **sp_changegroup grupo_test, nuevo2**

12. Conceder create table a todo el grupo_test

    - **grant create table to grupo_test**

13. Ver los permisos de todos los objetos creados

    - **sp_helpprotect gr_tabla | nuevo1 | nuevo2 | user1 | grupo_test**

1.9 Roles

1. Crear roles

   - **create role role1**
   - **create role role2**

2. Conceder permisos

   - **grant role role1 to login1**
   - **grant role role2 to login2**

3. Grant SA role a user1

   - **sp_role "grant", "sa_role", login1**
   - **sp_displaylogin login1**

4. Ahora conectar como user1 y crear un nuevo login. Qué ocurre? Resolver el problema !!

   - **use database_test**
   - **create proc test_sso_role**

```
if proc_role("sso_role") = 0

        begin

                print "Requerido SSO role"

                return

        end

else

        print "Perfecto :-) !!!"
```

   - **exec test_sso_role**
   - **set role "sso_role" off**
   - **exec test_sso_role**

1.10 Configurar el número de conexiones de usuario

1. Comprobar asignación de memoria vigente

   - **dbcc memusage**

2. Comprobar el valor por defecto del parámetro “User Environment” (25)

   - **sp_configure 'User Environment'**

3. Cambiar la configuración del parámetro (+15) y rebotar ASE

   - **sp_configure "user_connections", 40**

4. Comprobar nueva asignación de memoria vigente

   - **dbcc memusage**

   15 conexiones requieren 15 * (95 * 1024) / 2048 = 721 paginas

1.11 Calcular el tamaño de la memoria cache (datos y proc.)

- Heurística 1:
  total = actual + pags para conexiones + overhead

  overhead = 1.5% de las pags para conexiones

- Heurística 2:
  proc. cache = max. conexiones * tamaño del plan mayor * 1.25

- Heurística 3:
  min. proc. cache = nº de procs * tamaño medio de plan * 1.25

- Como aumentar la memoria:

  - **sp_configure "total memory", ?**

1.12 Named Caches

1. Crear una named cache para log de 2MB

   - **use master**
   - **sp_cacheconfig log_cache, "2M", logonly**
   - **reboot**

2. Reservar 512Kb de espacio para un buffer de intercambio (pool) de 4Kb por operación de E/S

   - **sp_poolconfig log_cache, "512K", "4K"**
   - **sp_poolconfig log_cache** → para ver si se ha creado correctamente

3. Poner la base de datos database_test en modo "single user" y actualizar las caches

   - **sp_dboption database_test, "single user", true**
   - **use database_test**
   - **checkpoint** → pasa los datos de las caches al disco de forma que la información almacenada sea consistente.

4. Enlazar la tabla de syslogs de database_test para que utilice la cache creada

   - **sp_bindcache log_cache, database_test, syslogs**

5. Poner database_test en modo multiusuario y hacer un checkpoint

   - **use master**
   - **sp_dboption database_test, "single user", false**
   - **use database_test**
   - **checkpoint**
   - **sp_helpcache log_cache**

6. Forzar la utilización de 4Kb para el log de operaciones de E/S

   - **use database_test**
   - **sp_logiosize "4K"**
   - **sp_logiosize**

1.13 Backups

1. Crear dos dispositivos de backup: uno para datos y otro para log

   - **sp_addumpdevice "disk", "dump_data", "/sybase/tests/dump_device.dat"**
   - **sp_addumpdevice "disk", "dump_log", "/sybase/tests/dump_device.log"**

2. Crear un tabla en database_test e introducir algún dato

   - **use database_test**
   - **create table dump_table (campo1 int, campo2 char(25))**
   - **insert dump_table values (1, "línea 1")**

3. Volcar la base de datos database_test en el dispositivo de backup

   - **dump database database_test to dump_data**

4. Modificar datos de dump_table y ver el contenido

   - **insert dump_table values (2, "línea 2")**
   - **select * from dump_table**

5. Cargar la base de datos y ver el contenido

   - **use master**
   - **load database database_test from dump_data**
   - **online database database_test**
   - **use database_test**
   - **select * from dump_table**

6. Insertar otra vez algún valor en la tabla

   - **use database_test**
   - **insert dump_table values (2, "línea 2")**

7. Volcar la base de datos

   - **dump database database_test to dump_data**

8. Modificar datos

- **use database_test**
- **insert dump_table values (3, "línea 3")**

9. Volcar el log de transacciones

- **dump tran database_test to dump_log**

10. Cargar la base de datos

- **use master**
- **load database database_test from dump_data**
- **online database database_test**
- **use database_test**
- **select * from dump_table**

11. Cargar el log

- **use master**
- **load tran database_test from dump_log**
- **online database database_test**
- **use database_test**
- **select * from dump_table**

1.14 Automatización de Backup

Procedimiento que realiza un backup del log automáticamente cuando quedan menos de 256 páginas libres.

- **create proc sp_pagesfree**
  @database varchar(30),
  @segment varchar(30),
  @espacio int,
  @status int
  as
  dump transaction @database with truncate_only
  print "LOG DUMP: '%1!' para '%2' volcado ('%3!' pages free)",
  @segment, @database

- Asociar el procedimiento al segmento de log

- **sp_addthreshold database_name, logsegment, 256, sp_pagesfree**

- Permisos de ejecución para el proc

- **grant exec on sp_sp_pagesfree to dbo**

1.15 Como recuperarse ante un fallo del master

1. Crear y ejecutar los dos procedimientos siguientes. Guardar los resultados en un archivo para futuras referencias. Estos dos procedimientos se utilizan para obtener los datos básicos necesarios de las tablas que gestionan el almacenamiento. Son imprescindibles si el master se corrompe.

   - **create proc sp_show_devices**
     as

     select low, high, status, name

     from master.dbo.sysdevices

     where cntrltype = 0

   - **grant execute on sp_show_devices to public**
   - **create proc sp_show_usages**
     as

     select dbid, lstart, size, vstart

     from master.dbo.sysusages

     order by vstart

   - **grant execute on sp_show_usages to public**

2. Guardar los resultados de las siguientes query's

   - **select * from sysdatabases**
   - **select * from sysloginroles**
   - **select * from syslogins**

3. Hacer un backup del master y del fichero de configuración de ASE.

   - sp_addumpdevice "disk", "master_dump", "/sybase/tests/master_dump_dev.dat"
   - **dump database master to master_dump**

Simulación de una situación de desastre:

- Shutdown de los servidores.
- mv /sybase/master /sybase/master.old (con esto nos cargamos el master)

---

4. Crear un master genérico de 17MB (debería ser de unos 30MB por lo menos)

   buildmaster –d /sybase/master –s 8704 (en paginas de 2K)

5. Arrancar el servidor en modo “master_recover” de forma que nos permita modificar las tablas del sistema. Modificar el fichero de arranque de ASE (RUN_nombremaq) añadiéndole la opción –m en la última línea.

   ./startserver –f /RUN_nombremaq

6. Redefinir las asignaciones de espacio en master. Reconstruir sysdevices y sysusages. ¡¡ MUCHO CUIDADO !!

7. Actualizar sysservers para definir un servidor de backup

   - **begin transaction**
   - **update sysservers**
   - **set srvnetname = “nombremaq_back”**
   - **where srvname = “SYB_BACKUP”**
   - **go**
   - **commit transaction**
   - **go**

8. Asegurarse de que el servidor de backup está arrancado.

9. Recuperar el último volcado del master.

   - **load database “master” from “master_dump”**

10. Reconfigurar el máximo número de dispositivos. Utilizar la copia de seguridad del archivo de configuración de ASE.

11. Reiniciar ASE en modo single_user (sin la –m en el archivo de arranque).

12. Verificar que todas las tablas del sistema coinciden con lo impreso.

13. Hacer shutdown y arrancar ASE en modo normal.

14. Hacer un backup del master, que ya está bien.