

COPIA DE SEGURIDAD COMPLETA

BACKUP

Para realizar una copia de seguridad completa de una base de datos, se deben realizar los siguientes comandos y en el orden siguiente orden:

<pre>BACKUP DATABASE nbd TO DISK = '\\path\copiabd.bak' WITH NAME = 'copial', INIT</pre>	1. Copia de la base de datos nbd en la ruta path con el nombre copiabd.bak, se marca con el nombre copia1 y se sobrescribe el archivo. Si se omite la opción INIT no se sobrescribe el archivo, opción por defecto.
<pre>BACKUP LOG nbd TO DISK = '\\path\copialog.bak' WITH NAME='copialog1', INIT</pre>	2. Copia del registro de transacciones. Al no indicar INIT tendremos tantas copias del registro como instrucciones se hayan ejecutado
<pre>BACKUP LOG nbd WITH TRUNCATE_ONLY</pre>	3. Realiza el truncado (vaciado) del fichero de transacciones.

COPIA DEL REGISTRO DE TRANSACCIONES (LOG)

BACKUP

Entre copia y copia de base de datos, se debe copiar el registro de transacciones. Para ello se ejecuta la siguiente instrucción:

<pre>BACKUP LOG nbd TO DISK = '\\path\copialog.bak' WITH NAME='copialog'</pre>	1. Copia del registro de transacciones. Al no indicar INIT tendremos tantas copias del registro como instrucciones se hayan ejecutado
--	---

RESTAURAR BASE DE DATOS

RESTORE

Para realizar recuperación completa de una base de datos, se deben realizar los siguientes comandos y en siguiente orden:

<pre>RESTORE DATABASE nbd FROM DISK='\\path\copialog.bak' WITH STANDBY='C:\standby.undo'</pre>	1. Restaurar la base de datos nbd desde el archivo: copiabd.bak. Con la opción STANDBY le indicamos que estamos probando si la restauración nos vale, ya que nos deja la base de datos en estado de sólo lectura, con lo que podremos acceder a ella. Si no lo hacemos así deberemos especificar la opción NORECOVERY, para indicarle que aún no hemos terminado con la recuperación.
<pre>RESTORE LOG nbd FROM DISK='\\path\copialog.bak' WITH STANDBY='\\path\standby.undo', FILE=1</pre>	2. Restaurar el registro de transacciones comenzando por el primero (FILE=1), hicieron después de la copia completa recuperada en la instrucción anterior. Realizar la comprobaciones necesarias, hasta determinar si nos vale la recuperación. Si especificamos la opción STANDBY, se debe usar NORECOVERY, para indicarle que aún no hemos terminado con la recuperación.

```
RESTORE LOG nbd
FROM DISK='\\path\copialog.bak'
WITH RECOVERY,
FILE=8
```

3. Restaurar el último registro con la opción RECOVERY (recuperación). De esta forma la base de datos se nos queda en modo escritura y lista para trabajar.

ESQUEMA TABLAS

A veces es necesario obtener una lista de los campos de una tabla así como el tipo y longitud. A continuación se muestra dos ejemplos:

```
select '@'+
    column_name+' '+
    UPPER(data_type)+' ('+
    LTRIM(STR(ISNULL(character_maximum_length,0))) +'), '
    from INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'ALARTDV'

select column_name+' = '+
    '@'+column_name+', '
    from INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'ALARTdv'
```

```
-----
-- eu_copiaBDatos -- 04.2008 -- Eurotronic -
-----
-- procedimiento que realiza la copia completa de la base de datos
-- @bdatos : indicar la base de datos
-- @path   : indicar la ruta destino de la copia
-----

DECLARE @bdatos VARCHAR(50)
DECLARE @path VARCHAR(250)
DECLARE @pathBD VARCHAR(250)
DECLARE @pathLG VARCHAR(250)
DECLARE @copia VARCHAR(50)

SET @bdatos = 'prueba' -- NOMBRE DE LA BASE DE DATOS
SET @path = 'c:\mssql\data\' -- RUTA DESTINO DE LA COPIA
SET @pathBD = @path + 'copiaBD_' + @bdatos + '.bak'
SET @pathLG = @path + 'copiaLG_' + @bdatos + '.bak'
SET @copia = GETDATE()

-- COMPROBAR EN N° DE COPIAS QUE HAY EN EL BACKUP DEL LOG
-- RESTORE HEADERONLY FROM DISK=@pathLG

-- COPIA COMPLETA DE LA BASE DE DATOS
BACKUP DATABASE @bdatos
    TO DISK = @pathBD
    WITH NAME = @copia ,
    INIT

-- COPIA DEL REGISTRO DE TRANSACCIONES
BACKUP LOG @bdatos
    TO DISK = @pathLG
    WITH NAME = @copia,
    INIT

-- TRUNCAMOS EL LOG PARA EMPEZAR UN NUEVO CICLO
BACKUP LOG @bdatos
    WITH TRUNCATE_ONLY
-----
-- RESTAURACION
-- RESTORE DATABASE PRUEBA
--     FROM DISK='C:\MSSQL\DATA\copiaBD_prueba.bak'
--     WITH NORECOVERY
```

```
-----  
-- eu_copiaLOG -- 04.2008 -- Eurotronic -  
-----  
-- procedimiento que realiza la copia del registro de transacciones  
-- periodicamente, por ejemplo cada hora.  
-- @bdatos : indicar la base de datos  
-- @path   : indicar la ruta destino de la copia  
-----  
  
DECLARE @bdatos VARCHAR(50)  
DECLARE @path VARCHAR(250)  
DECLARE @pathBD VARCHAR(250)  
DECLARE @pathLG VARCHAR(250)  
DECLARE @copia VARCHAR(50)  
  
SET @bdatos = 'prueba' -- NOMBRE DE LA BASE DE DATOS  
SET @path = 'c:\mssql\data\' -- RUTA DESTINO DE LA COPIA  
SET @pathBD = @path + 'copiaBD_' + @bdatos + '.bak'  
SET @pathLG = @path + 'copiaLG_' + @bdatos + '.bak'  
SET @copia = GETDATE()  
  
-- COMPROBAR EN N° DE COPIAS QUE HAY EN EL BACKUP DEL LOG  
-- RESTORE HEADERONLY FROM DISK=@pathLG  
  
-- COPIA DEL REGISTRO DE TRANSACCIONES  
BACKUP LOG @bdatos  
    TO DISK = @pathLG  
    WITH NAME = @copia
```

TRUNCAR Y REDUCIR EL LOG DE RPS

Desde la consola de SQL Server ir Administración – Agente –Trabajos

Añadir un nuevo trabajo tal como se indica

	Trabajo de pruebas de integridad para el plan ...	Mantenimiento de ...	Sí	Sí
	Truncar Log RPS	[Sin categoría (Lo...	Sí	Sí

Añadir 2 pasos:

Propiedades de Truncar Log RPS - (LOCAL)

General Paso Programaciones Notificaciones

Id.	Nombre del paso	Tipo	En caso de éxito	En caso de error
1	Truncar Log RPS	Secuencia d...	Ir al paso siguiente	Salir con error
2	Reducir Log RPS	Secuencia d...	Salir con éxito	Salir con error

Paso 1:

Modificar paso de trabajo - (LOCAL)\Truncar Log RPS

General Avanzadas

Nombre del paso: Truncar Log RPS

Tipo: Secuencia de comandos Transact-SQL (TSQL)

Base de datos: RPS

Comando: `BACKUP LOG RPS WITH TRUNCATE_ONLY`

Paso 2:

Modificar paso de trabajo - (LOCAL)\Truncar Log RPS

General Avanzadas

Nombre del paso: Reducir Log RPS

Tipo: Secuencia de comandos Transact-SQL (TSQL)

Base de datos: master

Comando: `USE [RPS] DBCC SHRINKFILE (N'RPS_LOG')`

Abrir

Por último programarlo 1 vez a la semana, justo después del chequeo de integridad.

USUARIOS DE BASE DE DATOS Y RPS

Requisitos para abrir RPS

Para un correcto funcionamiento de la aplicación y poder establecer una correcta política de seguridad en los datos, se deben cumplir los siguientes requisitos:

En SQL-Server:

En el apartado de "Inicios de sesión" deben existir las siguientes entradas:

Usuario	Contraseña	Idioma	Comentario
admusr	(se puede cambiar)	English	Usuario administrador de RPS
conexion	conexion	English	Usuario de conexion, no de trabajo.
emp	(se puede cambiar)	English	Usuario de base de datos asociado al perfil de empleado en la aplicación. Es da los permisos de acceso a las opciones de RPS y a la base de datos.
rps	rps	English	Usuario para poner en el origen ODBC

En RPS:

Crear los perfiles según los departamentos de la empresa, que luego se asignarán a los usuarios de la aplicación. Todos los perfiles creados deberán pertenecer a un usuario de base de datos, aunque sea el mismo: EMP.

Crear los usuarios de la aplicación y asociarlos al perfil (departamento) que pertenezcan. Por tanto la asociación será de la siguiente forma:

Usuario RPS ->	Perfil RPS ->	Usuario base de datos
Ejemplo: SCUESTA ->	EMP (Empleado) ->	EMP

En los Orígenes de Datos (ODBC):

Crear una entrada en los DSN de sistema que apunte al servidor SQL-Server, con un usuario de inicio de sesión válido, por ejemplo RPS. Dicho usuario, no es necesario que tenga acceso a la base de datos ya que solo se usa en la entrada ODBC para obtener el nombre del servidor SQL-Server. La validación se realiza por el usuario de base de datos asociado al usuario de RPS según el apartado anterior.

En el fichero INI:

Cambiar la entrada DSN= y poner el nombre del origen ODBC creado anteriormente
Cambiar la entrada DATABASE= y poner el nombre de la base de datos a la que queremos conectarnos.

Usuario de sólo consulta

En ciertos equipos puede ser necesario tener acceso a la base de datos para consultas mediante la conexión ODBC por aplicaciones externas a RPS, por ejemplo MS-Office, Crystal Reports, etc. Para ello se deben de cumplir los siguientes requisitos:

En SQL-Server:

En la Seguridad, crear un inicio de sesión con el nombre RPS, contraseña RPS y con permiso de acceso a la base de datos.

En la base de datos, crear una función (grupo de usuarios) con el nombre CONSULTA, con permiso de SELECT para todas las tablas de la base de datos.

Asociar a dicha función (grupo) el usuario RPS.

Para llevar a cabo todas las tareas anteriores se suministra a continuación un procedimiento T-SQL que las realiza.

En los Orígenes de Datos (ODBC):

Crear una entrada o sustituir la existente cambiando el usuario por RPS

```
-----
-- eu_usuarioConsulta -- 04.2008 -- Eurotronic --
-----

-- procedimiento que crea el usuario RPS en la bd
-- indicada con permisos de solo de SELECT

DECLARE @bdatos VARCHAR(250)-- nombre BD
DECLARE @usuario VARCHAR(250) -- nombre Usuario
DECLARE @password VARCHAR(250) -- contraseña
DECLARE @funcion VARCHAR(250) -- funcion de acceso
DECLARE @tabla SYSNAME -- nombre de la tabla o vista

SET @bdatos = 'ALMENSUR' -- nombre de la b.d.
SET @usuario = 'RPS' -- nombre del usuario
SET @password = 'RPS' -- contraseña
SET @funcion = 'CONSULTA' -- nombre del perfil de seguridad

-- nos posicionamos en la bd
EXEC ( 'USE ' + @bdatos )

-- 1. crear el usuario (inicio de sesión) en SQL Server
EXEC sp_addlogin @usuario, @password, @bdatos

-- 2. añadir funcion de acceso a la bd seleccionada
EXEC sp_addrole @funcion

-- 3. dar acceso al usuario a la base de datos
--EXEC sp_revokedbaccess @usuario
EXEC sp_grantdbaccess @usuario

-- 4. añadir el usuario a la funcion creada al bd seleccionada
EXEC sp_addrolemember @funcion, @usuario

-- 5. recorrer todas las tablas de la bd y asignar permiso de lectura
-- crea el cursor las tablas y vistas
DECLARE c_tablas CURSOR FOR
    SELECT table_name FROM INFORMATION_SCHEMA.TABLES

-- abre el cursor
OPEN c_tablas

-- situar en el primero
FETCH NEXT FROM c_tablas INTO @tabla

-- mientras no se llegue al final
WHILE @@FETCH_STATUS = 0
BEGIN

    -- asignar permiso SELECT a todas las tablas y vistas
    PRINT @TABLA

    EXEC ('GRANT SELECT ON ' + @tabla + ' TO ' + @funcion)

    --- pasar al siguiente
    FETCH NEXT FROM c_tablas INTO @tabla
END
CLOSE c_tablas
DEALLOCATE c_tablas

-- 5. fin
```

