



Copias de Seguridad de Bases de Datos

Contenido:

- 6.1.- Preparar el Ambiente de Trabajo
- 6.2.- Crear el Dispositivo de Almacenamiento
- 6.3.- Crear la Tabla de Prueba
- 6.4.- Realizar el Backup de la Base de Datos
- 6.5.- Realizar un Primer Backup Diferencial
- 6.6.- Realizar un Segundo Backup Diferencial
- 6.7.- Error en la Base de Datos

6.1.- Preparar el Ambiente de Trabajo

Para esta práctica de laboratorio crearemos una base de datos a la que llamaremos **PeruDEV**, en la cual crearemos una tabla y sobre ella haremos la demostración.

```
use master
go

create database perudev
go
```

La base de datos debe estar en modo de recuperación completa o de registro masivo.

```
alter database perudev
    set recovery FULL
go

select databasepropertyex( 'perudev','Recovery' )
go
```

6.2.- Crear el Dispositivo de Almacenamiento

Crear el dispositivo:

```
exec sp_addumpdevice 'disk', 'PD', 'C:\Backup\PDDevice.bak'
go
```

Verificar:

```
select * from master.dbo.sysdevices
go
```

6.3.- Crear la Tabla de Prueba

Crear la tabla de prueba:

```
create table perudev..test(  
    id Int Identity Primary Key,  
    dato varchar(30)  
)  
go
```

Cargar datos:

```
declare @k int  
set @k = 0  
while (@k < 100)  
begin  
    begin tran  
        insert into perudev..test(dato) values('Backup de Base de Datos')  
        commit tran  
        set @k = @k + 1  
    end  
end  
go
```

6.4.- Realizar el Backup de la Base de Batos

Supongamos que este backup se realiza el día domingo por la noche.

Realizar el backup de la base de datos:

```
backup database perudev  
to pd  
with  
    format,  
    name = 'BakBD',  
    description = 'Backup de la base de datos completa'  
go
```

Verifiquemos el backup:

```
restore headeronly from pd  
go
```

6.5.- Realizar un Primer Backup Diferencial

Supongamos que esto se realiza a las 10 pm del día lunes.

Primero cargamos nuevos datos:

```
declare @k int
set @k = 0
while (@k < 100)
begin
    begin tran
    insert into perudev..test(dato) values('SQL Server 2000')
    commit tran
    set @k = @k + 1
end
go
```

Realizamos el backup:

```
backup database perudev
to pd
with
    differential,
    name = 'BakDif01',
    description = 'Primer backup diferencial dela base de datos'
go
```

Verificar el backup:

```
restore headeronly from pd
go
```

6.6.- Realizar un Segundo Backup Diferencial

Supongamos que esto se realiza a las 10 pm del día martes.

Primero cargamos nuevos datos:

```
declare @k int
set @k = 0
while (@k < 100)
begin
    begin tran
        insert into perudev..test(dato) values('2do. Backup Diferencial')
        commit tran
        set @k = @k + 1
    end
end
go
```

Realizamos el backup:

```
backup database perudev
to pd
with
    differential,
    name = 'BakDif02',
    description = 'Segundo backup diferencial de la base de datos'
go
```

Verificar el backup:

```
restore headeronly from pd
go
```

6.7.- Error en la Base de Datos

Supongamos que se produce un error en la base de datos a las 11 am del día miércoles.

Simularemos actividad antes de las 11 am:

```
declare @k int
set @k = 0
while (@k < 100)
begin
    begin tran
    insert into perudev..test(dato) values('Que pasa con mis datos')
    commit tran
    set @k = @k + 1
end
go
```

En el momento que se produce un error en la base de datos, debemos hacer una copia de respaldo del log:

```
backup log perudev
to pd
with
    no_truncate,
    name = 'BakLog',
    description = 'Backup del log.'
go
```

Verificar la copia de respaldo:

```
restore headeronly from pd
go
```