http://www.sqlservercentral.com/articles/Administration/2733/
Printed 2007/11/28 04:09PM

Database Snapshots in SQL Server 2005

By Arvinder Khosla, 2006/11/26

Database Snapshots in SQL Server 2005

By Arvinder Singh Khosla and S.Srivathsani Murthy

Introduction

Database snapshots is a new feature added in SQL Server 2005.Database Snapshot can be used to create a read-only copy of the database. Database snapshot consists of the read-only static view of the database without including the uncommitted transactions. The uncommitted transactions are rolled back to make the database snapshot transactionally consistent.

Uses:

Database snapshots can be used for

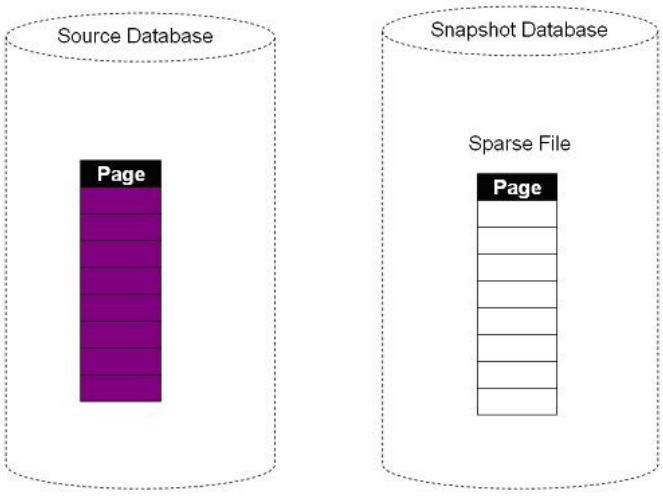
- Protecting your system from user or administrator error
- Offloading reporting
- Maintaining historical data
- System upgrades
- Materializing data on standby servers
- Recovering data

How Database Snapshots Work?

Database snapshots occur at the page level. Before a page of the source database is modified for the first time, the original page is copied from the source database to the snapshot. This process is called a copy-on-write operation. The snapshot stores the original page, preserving the data records as they existed when the snapshot was created.

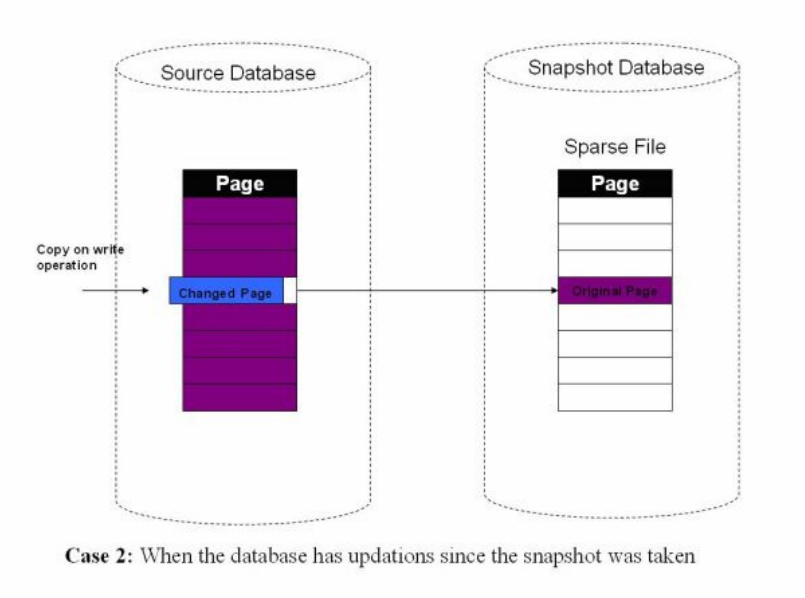
To store the copied original pages, the snapshot uses one or more sparse files. Initially, a sparse file is an essentially empty file that contains no user data and has not yet been allocated disk space for user data. As more and more pages are updated in the source database, the size of the file grows. When a snapshot is taken, the sparse file takes up little disk space. As the database is updated over time, however, a sparse file can grow into a very large file.
This is illustrated with the help of this diagram:

1. At first when the snapshot is created, the sparse file is empty.

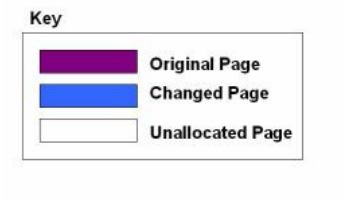


Case 1: When the database has no updations since the snapshot was taken

2. When the first update operation happens, the original data is copied into the sparse file. The database engine then updates the data in the source database.



Case 2: When the database has updations since the snapshot was taken



When the first update operation happens, the original data is copied into the sparse file. The database engine then updates the data in the source database.
During the read operation on the snapshot, the original data is read from the source database. For the data which has been changed, the snapshots read from the sparse file.

Advantages

1. The user can create as many snapshots as he/she wants quickly in no amount of time. The user can schedule to take snapshots every hour. This will be useful in auditing scenarios
2. The snapshots can be used in restore operations.
3. The corrupted or deleted data can be recovered from the snapshot to repair the primary database.
4. In case of user error, the administrator can revert back to the snapshot taken just before the error.

Disadvantages

1. Database snapshots are available only in the SQL Server 2005 enterprise edition.
2. Database snapshots are dependent on the primary database. If the primary database goes offline, then the snapshots are not accessible.
3. Performance of the source database is reduced, due to increased I/O on the source database resulting from a copy-on-write operation to the snapshot every time a page is updated.
4. Full-text indexing is not supported in database snapshots.
5. If the data changes rapidly, the snapshot might run out of disk space.

Creating a Database Snapshot-TSQL

In the **PUBS** database we have 3 data files under 2 file groups:

1. **Primary File group** contains Pubs and pubs_logicalname
2. **Pubs_data filegroup (Secondary Filegroup)** contains Pubs_2

The statement below creates a snapshot on the **PUBS** database

```
CREATE DATABASE pubs_Snapshot1800 ON
(Name=pubs,Filename='C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs_data1800.ss'),
(Name=pubs_logicalname,Filename='C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs_logicalname_data1800.ss'),
(NAME = pubs_2,
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs2_data1800.ss')
AS SNAPSHOT OF pubs
```

In pubs database, there are three files. The user should include all the three files while taking the snapshot. The database is a combination of data present in all three files. So we need to mention the all the three logical file names, so that all the data from these files will get stored in the location 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\filename_with_timestamp.ss'. In the example given above,

- pubs_Snapshot1200 --> Snapshot Name
- pubs --> Logical File Name

- C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\Northwind_data1000.ss --> Physical File Path

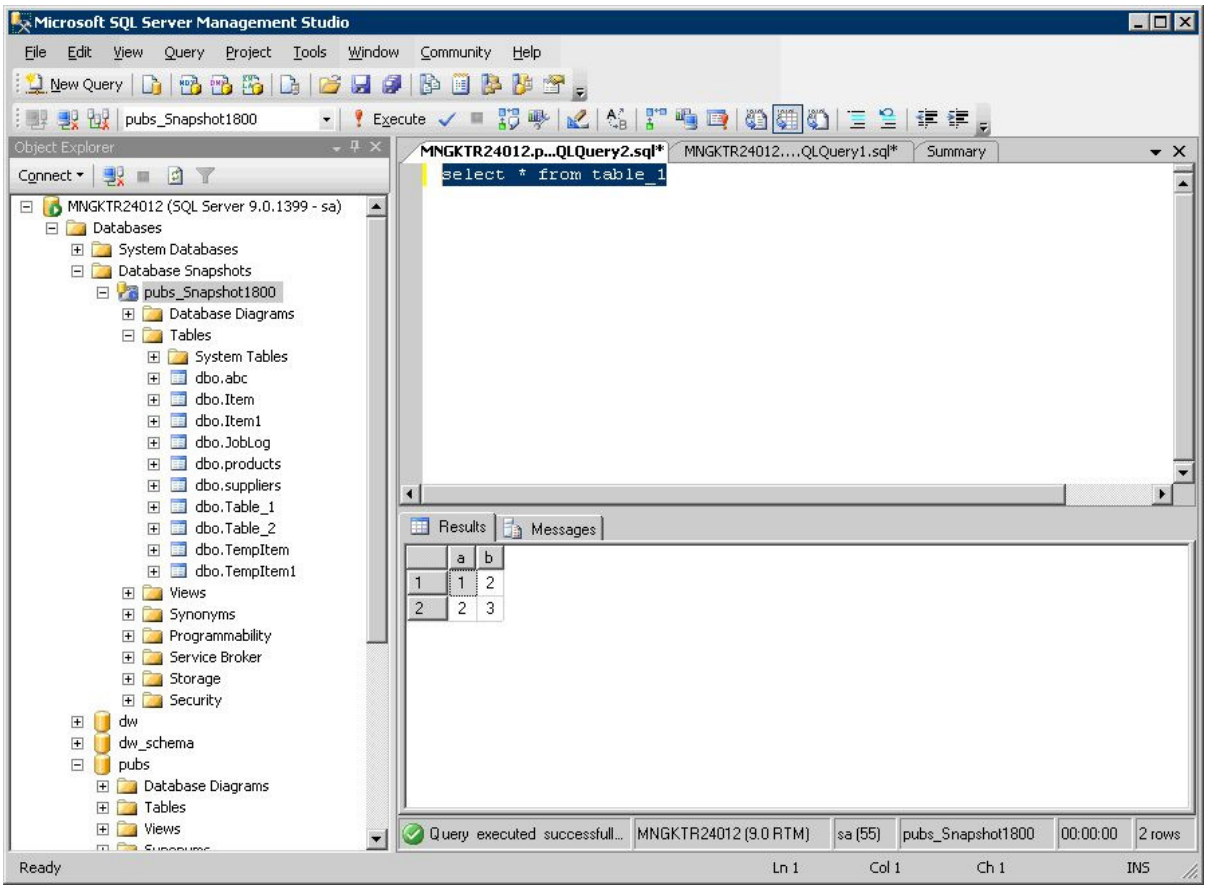
Suppose if the user wants to create the snapshot at 2400 hrs then he has to just change the name of the database and the physical file name.

```
CREATE DATABASE pubs_Snapshot2400 ON
(Name=pubs,Filename='C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs_data2400.ss'),
(Name=pubs_logicalname,Filename='C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs_logicalname_data2400.ss'),
(NAME = pubs_2,
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\pubs2_data2400.ss')
```

The snapshot can be used just like the database. The user has to remember that the snapshot is a read-only copy of the database. So he/she cannot issue the update, delete or insert commands.

Viewing Snapshots from SQL Server Management Studio

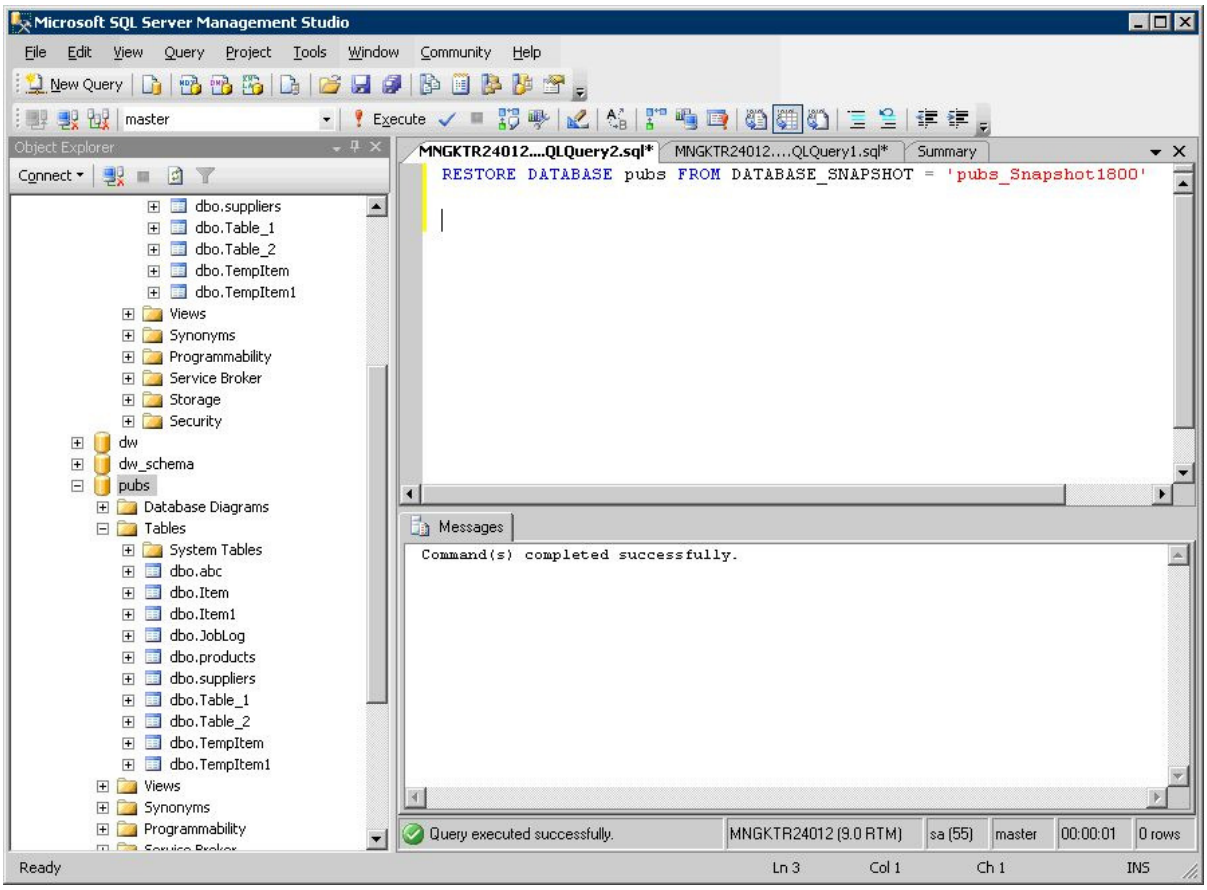
1. Expand Databases.
2. Expand Database Snapshots.
3. Select the snapshot created in the above section



Restoring from the Database Snapshot

Consider the database Pubs. Suppose the user deletes dbo.table_1 from the Pubs database. We have to recover it from the database snapshot by issuing the following command.

```
Restore database pubs from database_snapshot = 'pubs_Snapshot1800'
```



Dropping the Snapshot

```
Drop Database 'Snapshot_Name'
```

Frequently Asked Questions on Snapshots

Question 1. How is Snapshot different from the full backup of the database?
Answer 1. Differences are as follows:

1. In case of backup, we have to restore it somewhere to actually use it. The user cannot query the backup file. On the other hand snapshots are static read only data at that point of time. So it can be used just like any other database.
2. Another advantage or difference that snapshots have over backups is that backups occupy lot of space.
   For example: Suppose if there is a database in which only few updates takes place. Then definitely snapshot is a better option because less data has to be stored in the sparse file.

If we remember that in case of snapshots, only the pages which are updated are stored in the sparse file (refer above). When we fire the select query, the snapshots will read the pages which are not changed from the source database and the changed ones are read from the sparse file. Thus snapshot proves more advantageous than full backups in these scenarios.

Question 2. Is ss the default extension of the data files?
Answer 2. The default extension is not ss. It can be anything

Question 3. When to use Snapshot and when to use Full backups? And are differential backups influenced / dependent on Snapshot?
Answer 3.
Backups and Snapshots are two different concepts. Snapshots are definitely NOT a replacement for backups. Snapshots should be used for auditing or reporting purposes. Recovery operations can be done with the help of snapshots. But it is not the primary resource. Snapshots provide a read only view of the database at that point of time. It does not include uncommitted transactions. When we take backups, the uncommitted transactions are also included in the backups.
Differential backups are not dependent on snapshots. Differential backup is the differential of the last full backup and the current state of the database.

Got more questions?
Please mail us (Arvinder Singh Khosla or Srivathsani Murthy)