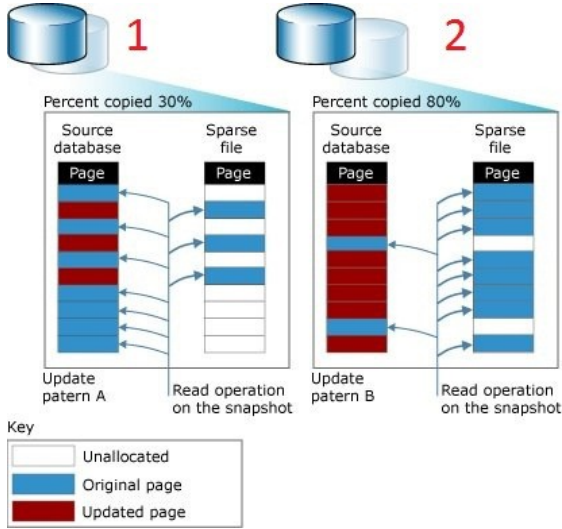


SQL Server Veri Tabanı Snapshot Alma

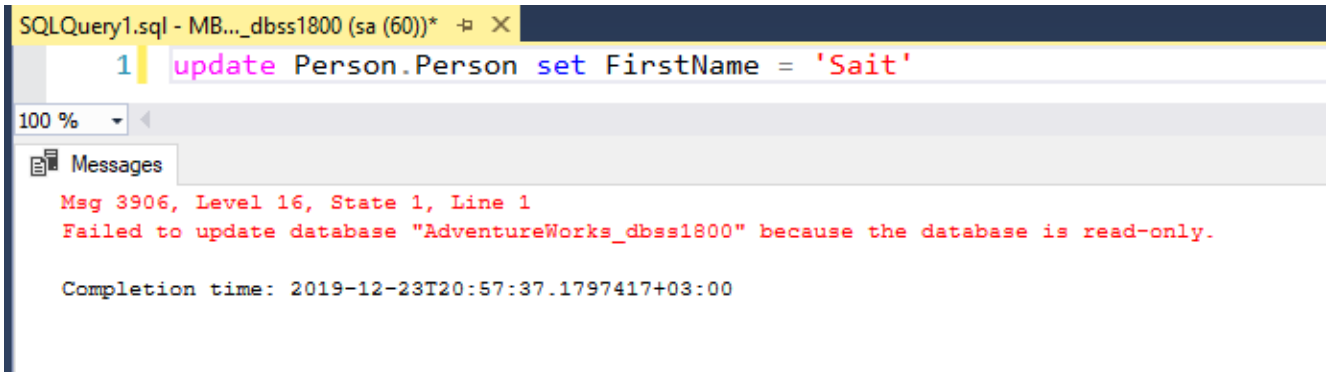


Snapshot ifadesi genel olarak anlık sistemin anlık fotoğrafı olarak tanımlanabilir. Snapshot alma amacı, sistemin anlık görüntüsünü alıp sonradan sistemi önceki haline döndürmek gerekmesi durumunda döndüre bilmektir. Snapshot alındıktan sonra anlık görüntü için bir fark dosyası oluşturulur. Bu andan itibaren veri tabanında bir değişiklik yapıldığında verinin snapshot alındığı andaki hali bu fark dosyasına yazılır ve alınan bu snapshot veri tabanına bir select sorgusu geldiğinde veri değişmemişse orijinal veri tabanından, veri değişti ise önceki halini tutan snapshot veri tabanından sorgu çekilir.

SQL Server Management Studio görsel olarak snapshot alma işlemini desteklemediğinden işlemlerin T-SQL kodu ile yapılması gerekmektedir. Oluşturulan snapshotlar SSMS'de "Database Snapshots" bölümünü altında yer alır.

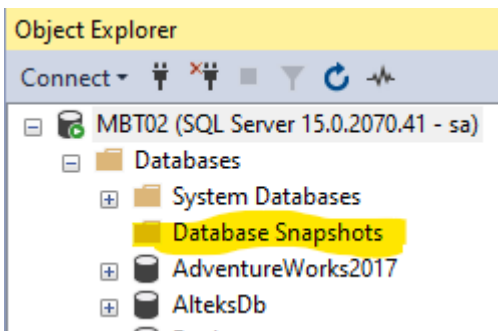
Oluşturulan bu snapshot veri tabanları salt okunur veri

tabanlarıdır ve sadece select cümleliğini desteklerler.



```
SQLQuery1.sql - MB..._dbss1800 (sa (60))*  
1 | update Person.Person set FirstName = 'Sait'  
100 %  
Messages  
Msg 3906, Level 16, State 1, Line 1  
Failed to update database "AdventureWorks_dbss1800" because the database is read-only.  
Completion time: 2019-12-23T20:57:37.1797417+03:00
```

Snapshot veri tabanı üzerinde UPDATE işlemi ve alınan hata



SSMS Database Snapshots

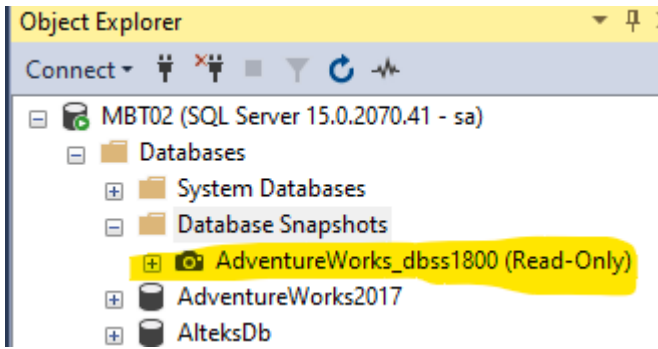
Snapshot almak için yazılması gereken T-SQL kodu aşağıdaki gibidir.

```
CREATE DATABASE snapshot_olarak_olusturulacak_dbAd  
ON  
(  
NAME =Mantıksal_dosya_ad,  
FILENAME ='olusturulacak_dosya_tam_ad'  
)
```

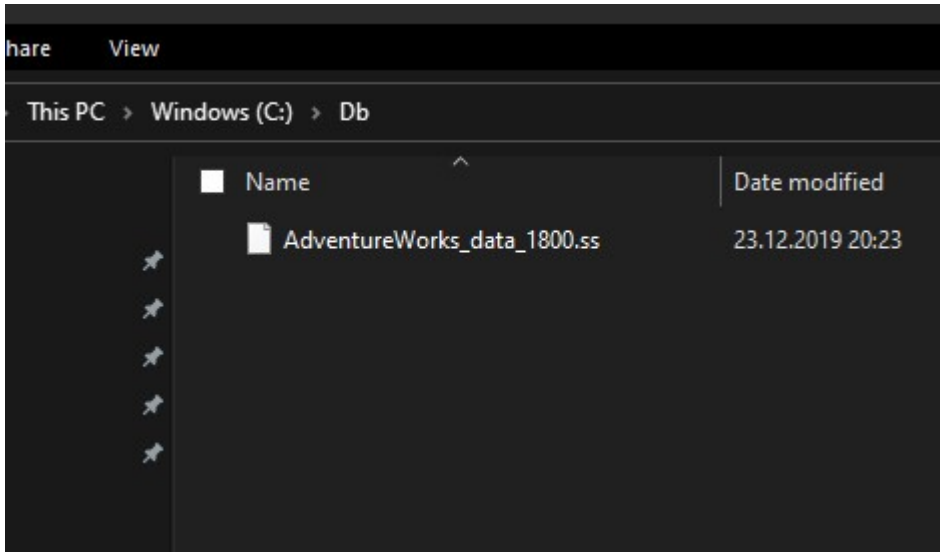
AS SNAPSHOT OF kaynak_dbAd

Aşağıdaki kodda da AdventureWorks veri tabanının bir snapshotunun alınmasını görüyoruz.

```
CREATE DATABASE AdventureWorks_dbss1800 ON  
( NAME = AdventureWorks2017,  
FILENAME =  
'C:\Db\AdventureWorks_data_1800.ss' )  
AS SNAPSHOT OF AdventureWorks2017;  
GO
```



Snapshot Oluşan Veri Tabanı

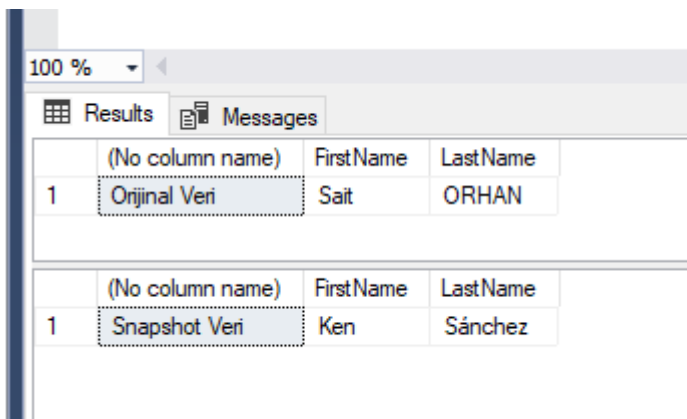


Snapshot İle Oluşan Fark Dosyası

Aşağıda snapshot alındıktan sonra değiştirilen verinin orijinal veri tabanından ve aynı verinin snapshot veri tabanından sorgulanması ve sonucunu görüyoruz.

```
--Snapshot alındıktan sonra değiştirilen veri sorgusu  
select 'Orijinal Veri', FirstName, LastName from  
AdventureWorks2017.[Person].[Person] where BusinessEntityID =  
1
```

```
--Snapshot alındıktan sonra değiştirilen verinin snapshot veri  
tabanından sorgusu  
select 'Snapshot Veri', FirstName, LastName from  
AdventureWorks_dbss1800.[Person].[Person] where  
BusinessEntityID = 1
```



	(No column name)	FirstName	LastName
1	Orijinal Veri	Sait	ORHAN

	(No column name)	FirstName	LastName
1	Snapshot Veri	Ken	Sánchez

Sistemde Var Olan Snapshotları Görmek için sys.databases kataloğundan **source_database_id** değeri NULL olmayan veri tabanları sorgulanır.

```
select * from sys.databases where source_database_id is not  
null
```

Veri tabanının snapshot zamanında ki durumuna dönmek için de aşağıdaki kodu kullanabiliriz. Snapshot verisine dönebilmek

için veri tabanına kullanıcıların bağlı olmaması gerekmektedir. Kullanıcıların bağlı olması durumunda geri dönme sırasında aşağıdaki hata alınır.

```
100 %
Messages
Msg 5070, Level 16, State 2, Line 3
Database state cannot be changed while other users are using the database 'AdventureWorks2017'
Msg 3013, Level 16, State 1, Line 3
RESTORE DATABASE is terminating abnormally.

Completion time: 2019-12-23T20:48:26.4463731+03:00
```

USE master;

-- AdventureWorks2017 veri tabanını AdventureWorks_dbss1800 zamanına döndürme

```
RESTORE DATABASE AdventureWorks2017 from
DATABASE_SNAPSHOT = 'AdventureWorks_dbss1800';
GO
```

Geri dönüş olduktan sonra az önce değiştirdiğimiz veriyi asıl veri tabanından sorgulayıp sonucu görelim.

```
8
9 select 'Orijinal Veri', FirstName, LastName
10 from AdventureWorks2017.[Person].[Person]
11 where BusinessEntityID = 1
```

(No column name)	FirstName	LastName
Orijinal Veri	Ken	Sánchez

Yazımızı bitirmeden snapshot veri tabanları konusunda söylememiz gereken son bir konu da, snapshotların kendi başlarına bir veri tabanı olmadıkları ve sistemde yaşamaları kaynak veri tabanlarınınin sistemde yaşıyor olması gerekliliğidir.