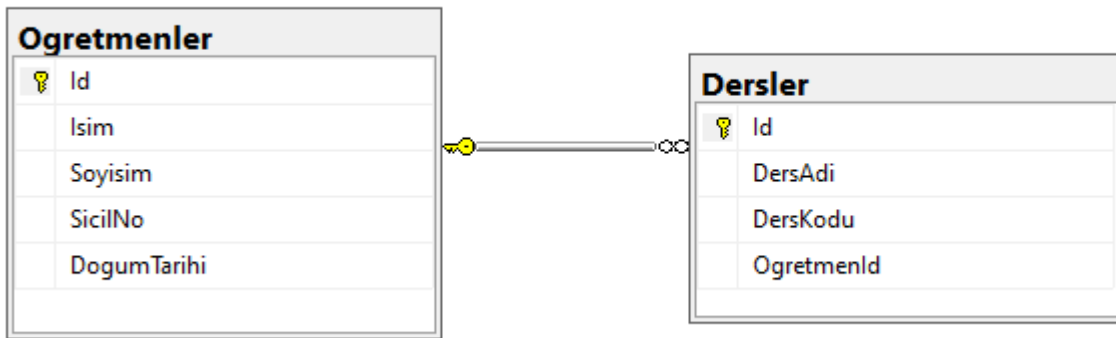


# SQL Server DML Komutlarına Hızlı Bakış

DML (Data Manipulation Language) komutları, tabloya veri ekleme, tablodan veri güncelleme, silme ve veri sorgulama işlemlerinde kullanılan komutlardır. Bu komutlar, yaptıkları işlemlerin baş harflerinin alınmasından oluşan CRUD kelimesi ile de tabir edilir. Bu komutlar;

- Veri Girişi (INSERT)
- Okuma (Select)
- Güncelleme (UPDATE)
- Silme (DELETE)

Makalemiz boyunca yapacağımız örnekler aşağıdaki tablo yapısı üzerinden ilerleyecektir.



## Veri Girişi (INSERT)

CRUD kelimesinin ilk harfinde geçen CREATE işlemidir. INSERT

komutu ile tabloya veri giriři saęlanır. Temel kullanımı ařaęıdaki gibidir.

```
INSERT INTO Tablo_Ad(col1, col2,....., coln) VALUES(deger1, deger2, ..., degern)
```

Tablo adından sonra gelen parantez ierinde veri ekleme iřleminde deęer atanacak kolonların adları ve VALUES anahtar kelimesinden sonra gelen parantezde ise bu kolonların sırası ile deęerleri belirtilir.

Ařaęıda retmenler tablosuna bazı veri giriři rneklerini inceleyelim.

```
INSERT INTO Ogretmenler(Isim, Soyisim, SicilNo, DogumTarihi) VALUES('Sait', 'ORHAN', '123456', '1989-06-28')
```

Bu rneęimizde kolonlara sırası ile ařaęıdaki deęerler atanmıřtır.

Isim	Sait
SoyIsim	ORHAN
SicilNo	123456
DogumTarih	28.06.1989

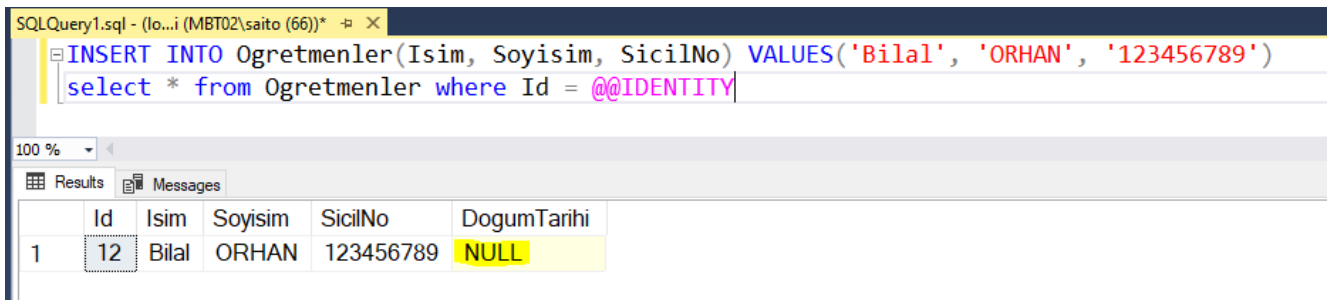
Bu rnekte olduęu gibi btn kolonlara deęer atanacak ise tablo isminden sonra parantez ierisinde kolon isimlerini belirtmeye gerek yoktur. Byle bir kullanımda da VALUES anahtar deęerleri kolonların tablodaki sıralarına gre atanacaktır. rnek kullanımı ařaęıdaki gibidir.

```
INSERT INTO Ogretmenler VALUES('Sait', 'ORHAN', '123456', '1989-06-28')
```

Veri ekleme işleminde null değer alabilen kolonlara değer atanmadan da işlem yapılabilir. Örneğin aşağıdaki sorguda null değer alabilen DogumTarih kolonuna değer atanmamış şekilde tabloya kayıt girilmektedir.

```
INSERT INTO Ogretmenler(Isim, Soyisim, SicilNo) VALUES('Bilal', 'ORHAN', '123456789')
```

Bu sorgu kaydedilen satırın görüntüsü aşağıdaki gibi olur.



```
SQLQuery1.sql - (lo...i (MBT02\saito (66))*) ×
```

```
INSERT INTO Ogretmenler(Isim, Soyisim, SicilNo) VALUES('Bilal', 'ORHAN', '123456789')  
select * from Ogretmenler where Id = @@IDENTITY
```

100 %

Results Messages

	Id	Isim	Soyisim	SicilNo	DogumTarihi
1	12	Bilal	ORHAN	123456789	NULL

**@@IDENTITY** değeri tabloda otomatik olarak artan kolonun son atanan değerini verir.

Aynı anda birden fazla satır eklemek için de VALUES anahtar kelimesinden sonra satırları virgül ile ayrılarak eklenebilir.

```
INSERT INTO Ogretmenler(Isim, Soyisim, SicilNo) VALUES('Çetin', 'Güneş', '123456785'),('Ahmet', 'Yıldız', '123456780')
```

## Veri Güncelleme (UPDATE)

CRUD kelimesindeki U harfinde belirtilen UPDATE işlemi tablodaki verilerin güncellenmesini sağlar. Tablodaki belli kolonların bütün değerlerini değiştirebileceği gibi WHERE anahtar kelimesi ile sadece belirtilen şartları sağlayan satırların değerleri de değiştirilebilir. Temel kullanım şekli aşağıdaki gibidir.

```
UPDATE Tablo_Ad SET col1 = deger1, col2 = deger2,... coln = degern WHERE sart
```

UPDATE komutunun kullanımında WHERE ile sadece değeri değiştirilmek istenen kolonlara ait şartlar verilmelidir. Herhangi bir şart belirtilmeden komut çalıştırılırsa tablodaki bütün değerleri değiştirir.

Aşağıdaki örnekte sadece Id değeri 7 olan satırın isim ve soy isim alanları güncelleniyor.

```
UPDATE Ogretmenler SET Isim = 'Mustafa', Soyisim = 'Yıldırım'  
WHERE Id = 7
```

Aşağıdaki komutta ise öğretmenler tablosundaki bütün isimler ve soy isimler Mustafa ve Yıldırım olarak değiştirilmektedir.

```
UPDATE Ogretmenler SET Isim = 'Mustafa', Soyisim = 'Yıldırım'
```

## Veri Silme (DELETE)

CRUD kelimesindeki D harfinde belirtilen DELETE işlemi tablodaki verilerin silinmesini sağlar. Tablodaki belli tüm satırlar silinebileceği gibi WHERE anahtar kelimesi ile sadece belirtilen şartları sağlayan satırlar da silinebilir. Temel kullanım şekli aşağıdaki gibidir.

```
DELETE Tablo_Ad WHERE sart
```

DELETE komutunun kullanımında WHERE ile sadece silinmek istenen satırlara ait şartlar verilmelidir. Herhangi bir şart belirtilmeden komut çalıştırılırsa tablodaki bütün satırları.

Aşağıdaki örnekte sadece Id değeri 7 olan satır siliniyor.

```
DELETE Ogretmenler WHERE Id = 7
```

Aşağıdaki kullanımda ise öğretmenler tablosunun bütün içeriği silinmektedir.

```
DELETE Ogretmenler
```

## Veri Sorgulama (SELECT)

CRUD kelimesinde geçen R (READ) harfi ile ifade edilen işlemdir. Tablodan veri çekmek için kullanılır. Temel kullanımı aşağıdaki gibidir.

```
SELECT column1, column2, ...  
FROM tablo_ad  
WHERE sartlar
```

SELECT sorgusunda WHERE anahtar kelimesi ile bir şart verilmezse tablodan istenen kolonların bütün verilerini getirir. Örneğin aşağıdaki sorgu ile öğretmenler tablosunda yer alan bütün veriler listelenir.

**SELECT sorgusunda bütün kolonlar alınacak ise SELECT ifadesinden sonra bütün kolonları yazmak yerine \* işareti konarak da bütün kolonlar getirilebilir.**

```
SELECT * FROM Ogretmenler
```

veya \* yerine aşağıdaki gibi bütün kolonların adlarını yazmak da aynı sonucu getirecektir.

```
SELECT Id, Isim, Soyisim, SicilNo, DogumTarihi FROM  
Ogretmenler
```

SQLQuery1.sql - (lo...i (MBT02\saito (65))\*

```
SELECT * FROM Ogretmenler
```

100 %

Results Messages

	Id	Isim	Soyisim	SicilNo	DogumTarihi
1	7	Mustafa	Yıldırım	854	1985-12-12
2	8	Sait	ORHAN	123456	2020-06-28
3	11	Ömer	ORHAN	1234568	2020-06-28
4	12	Bilal	ORHAN	123456789	NULL
5	15	Çetin	Güneş	123456785	NULL

SELECT işleminde bütün kolonlar alınabileceği gibi sadece ihtiyaç duyulan kolonları ihtiyaç duyulan sırada da alınması mümkündür. Örneğin aşağıdaki sorguda öğretmenlerin isim, soy isim, doğum tarihi ve son kolon olarak da tabloda olmamasına rağmen formül yazılarak anlık hesaplanan yaşları kolonu listelenmiştir.

```
SELECT Isim, Soyisim, DogumTarihi, DATEDIFF(YEAR, DogumTarihi, GETDATE()) Yas FROM Ogretmenler
```

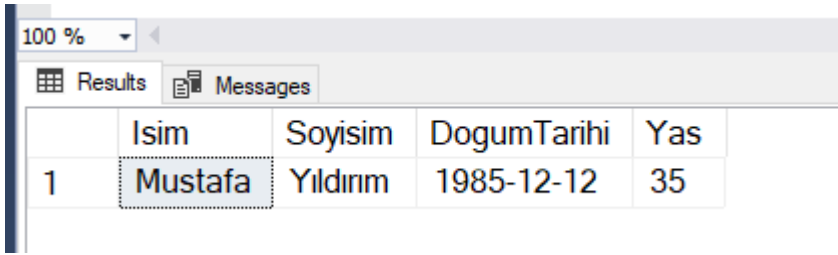
100 %

Results Messages

	Isim	Soyisim	DogumTarihi	Yas
1	Mustafa	Yıldırım	1985-12-12	35
2	Sait	ORHAN	1989-06-28	31
3	Ömer	ORHAN	2017-06-28	3
4	Bilal	ORHAN	NULL	NULL
5	Çetin	Güneş	NULL	NULL

Aşağıdaki sorguda da aynı sorguyu sadece Id değeri 7 olan öğretmen için sorgulanıyor.

```
SELECT Isim, Soyisim, DogumTarihi, DATEDIFF(YEAR, DogumTarihi, GETDATE()) Yas FROM Ogretmenler WHERE Id = 7
```



	Isim	Soyisim	DogumTarihi	Yas
1	Mustafa	Yıldırım	1985-12-12	35

## Sorgularda Birden Fazla Şartın Birleştirilmesi

Bazı sorgularda WHERE anahtar kelimesi ile verilen şartlarda birden fazla şart olabilmektedir. Bu durumda OR veya AND bağlaçları ile şartlar birleştirilir.

OR	Veya anlamındadır. OR ile bağlanan şartlardan en az biri doğru olduğunda şart sağlanmış olur.
AND	Ve anlamındadır. AND ile bağlanan şartların tümünün doğru olması durumunda şart sağlanmış olur.

Aşağıdaki sorguda Id değeri 7 olan veya sicil numarası 123456789 olan öğretmenler listelenmektedir.

```
SELECT * FROM Ogretmenler WHERE Id = 7 OR SicilNo = '123456789'
```



	Id	Isim	Soyisim	SiciNo	DogumTarihi
1	7	Mustafa	Yıldırım	854	1985-12-12
2	12	Bilal	ORHAN	123456789	NULL

---

## SQL Server Collations

SQL Server Collection ayarları verilerin karşılaştırma kuralları, büyük/küçük harf duyarlılığı ve aksanlı harflerin duyarlılık işlemlerinde kullanılan ayarlardır. Collection ayarlarından etkilenen veri türleri, metinsel ifade içeren char ve varchar veri türleridir.

SQL Server kurulumunda, yeni veri tabanı oluşturmada veya bir veri tabanı yedekten dönüldüğünde üzerinde çalışılacak verilerin büyük/küçük harf duyarlılığı, aksan duyarlılığı gibi durumlara toleransını bilmek sistemin hatasız çalışması üzerinde önemli bir şarttır. Bu durumda sisteme tanımlı collection ayarlarını bilmek sistemi ayarlama kolaylık sağlayacaktır. `sys.fn_helpcollations()` fonksiyonu ile sistemde yüklü bütün collection listesine ulaşılabilir.

SQLQuery1.sql - (lo...r (MBT02\saito (55))\* -# X

```
select * from sys.fn_helpcollations()
```

100 %

Results Messages

	name	description
1	Albanian_BIN	Albanian, binary sort
2	Albanian_BIN2	Albanian, binary code point comparison sort
3	Albanian_CI_AI	Albanian, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
4	Albanian_CI_AI_WS	Albanian, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
5	Albanian_CI_AI_KS	Albanian, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
6	Albanian_CI_AI_KS_WS	Albanian, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
7	Albanian_CI_AS	Albanian, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
8	Albanian_CI_AS_WS	Albanian, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
9	Albanian_CI_AS_KS	Albanian, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
10	Albanian_CI_AS_KS_WS	Albanian, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
11	Albanian_CS_AI	Albanian, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
12	Albanian_CS_AI_WS	Albanian, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
13	Albanian_CS_AI_KS	Albanian, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
14	Albanian_CS_AI_KS_WS	Albanian, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive

SQL Server, veri tabanı, kolon veya bir sorgu için collection ataması yapıldığında bu işlem veriler üzerinde yapılan sorgulama sonuçlarını değiştirir. Örneğin; Metinsel ifadelerin karşılaştırma, sıralama ve gruplama işlemleri bu ayardan etkilenen temel işlemlerdir.

Collection, verilerin kaydedilirken tutulduğu bit düzeyinde bilginin nasıl tutulacağını ve metinlerin sıralama ve karşılaştırma ayarlarını belirler. Bu ayarlar veri tabanı düzeyinde belirtilenebileceği gibi her kolona ayrı collection'da atanabilir. Collection'nın bu özelliğinden dolayı farklı collection ayarına sahip iki veri tabanı üzerinde çalıştırılan aynı sorguların sonuçları farklılık gösterebilir. Böyle bir durumla karşılaşmamak için mümkün olduğu sürece kurum içerisinde kullanılacak collection ile ilgili bir standart geliştirilmelidir.

SQL Server collection ayarlanırken kullanılan özellikler aşağıdaki tabloda görüldüğü gibidir;

Case-sensitive (_CS)	Büyük/küçük harf ayrımını yapar. Bu seçenek seçili iken küçük harfler büyük harflerden önce sıralanır. Bu seçenek seçili değil iken büyük / küçük harfler karşılaştırma ve sıralamada eşit sayılır. Seçili olmayan hali _CI ile ifade edilir.
Accent-sensitive (_AS)	Aksanlı ve aksansız harfleri ayırt eder. Bu seçenek seçili iken 'a' ve 'â' harfleri eşit sayılmaz. Bu seçenek seçili değil iken ise karşılaştırma ve sıralama işlemlerinde 'a' ve 'â' gibi harfler eşit sayılır. Seçili olmayan hali _AI ile ifade edilir.
Kana-sensitive (_KS)	Hiragana ve Katakana Japon karakterlerini ayırt eder.
Width-sensitive (_WS)	Özellikle matematiksel ifade içeren metinlerde tam ve yarım yazılan karakterleri ayır eder. Örneğin; Seçili iken: <b>E=mc<sup>2</sup></b> ve <b>E=mc2</b> ifadeleri eşit sayılmazken, seçili olmadığı durumda da bu iki ifade eşit kabul edilir.
Variation-selector-sensitive (_VSS)	Japoncaya özgü bazı karakterlerin ayırt edilmesini sağlar.

Binary (_BIN)	SQL Server tablolarındaki verileri, her karakter için tanımlanan bit kalıplarına göre sıralar ve karşılaştırır. İkili sıralama düzeni büyük / küçük harfe ve aksan yapısına duyarlıdır. Aynı zamanda en hızlı şekilde sıralamayı sağlar.
Binary-code point (_BIN2) <sup>1</sup>	Unicode verileri için Unicode kod noktalarını temel alarak SQL Server tablolarındaki verileri sıralar ve karşılaştırır. Unicode olmayan veriler için, İkili kod noktası ikili türlerle aynı karşılaştırmaları kullanır.
UTF-8 (_UTF8)	UTF-8 kodlu verilerin SQL Server üzerinde saklanmasını sağlar.

## Collation Seviyeleri

SQL Server üzerinde dört seviyede collation ayarı tutulabilmektedir. Bu seviyeler;

- Sunucu seviyesinde
- Veri tabanı seviyesinde
- Kolon seviyesinde
- Sorgu seviyesinde

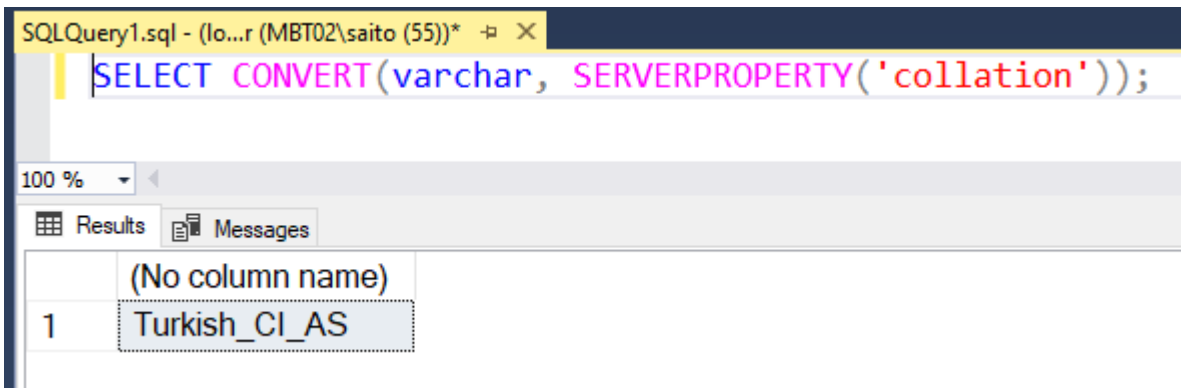
## Sunucu Seviyesinde

SQL Server kurulumu sırasında seçilir ve diğer bütün veri tabanı ve kullanıcıların varsayılan collection ayarı olur.

Kurulumdan sonra bu ayar değiştirilmek istenirse öncelikle bütün veri tabanlarının nesnelere ve verilerini dışarı aktardıktan sonra master veri tabanının rebuild işlemine tabi tutulması gerekmektedir. master veri tabanı rebuild edildikten sonra veri tabanları ve veriler tekrardan içeri aktarılır. Bütün bir sunucu seviyesinde collection ayarını değiştirmek yerine veri tabanı oluştururken veri tabanı seviyesinde de collection ayarı seçilebilir.

```
SELECT CONVERT(varchar, SERVERPROPERTY('collation'));
```

Yukarıdaki kod ile sunucunun mevcut collection ayarı sorgulanabilir.



## Sunucu Colletion Ayarının Değiştirilmesi

Sunucu colletion ayarını değiştirmek için aşağıdaki adımlar izlenir;

1. Veri tabanları ve nesnelerini yeniden oluşturmak için gerekli scriptler alınır.
2. Veri tabanlarındaki bütün veriler dışarı alınır. (export)
3. Bütün kullanıcı veri tabanları silinir.
4. master veri tabanı yeni collection ayarına göre rebuild edilir. Bu işlem için aşağıdaki setup komutu ile yapılabilir.

```
Setup /QUIET /ACTION=REBUILDDATABASE  
/INSTANCENAME=InstanceName  
/SQLSYSADMINACCOUNTS=accounts /[ SAPWD= StrongPassword ]  
/SQLCOLLATION=CollationName
```

5. Veri tabanları ve nesnelere scriptler yardımı ile yeniden oluşturulur.

6. Veriler tekrar içeri alınır. (import)

## **Veri Tabanı Seviyesinde Collection**

Veri tabanı oluşturulurken seçilir. Veri tabanı oluşturma sırasında seçilmezse varsayılan olarak sunucu seviyesinde seçilen collection seçilir. Veri tabanı oluşturulma veya değiştirilme işlemlerinde **COLLATE** komutu ile seçilebilir.

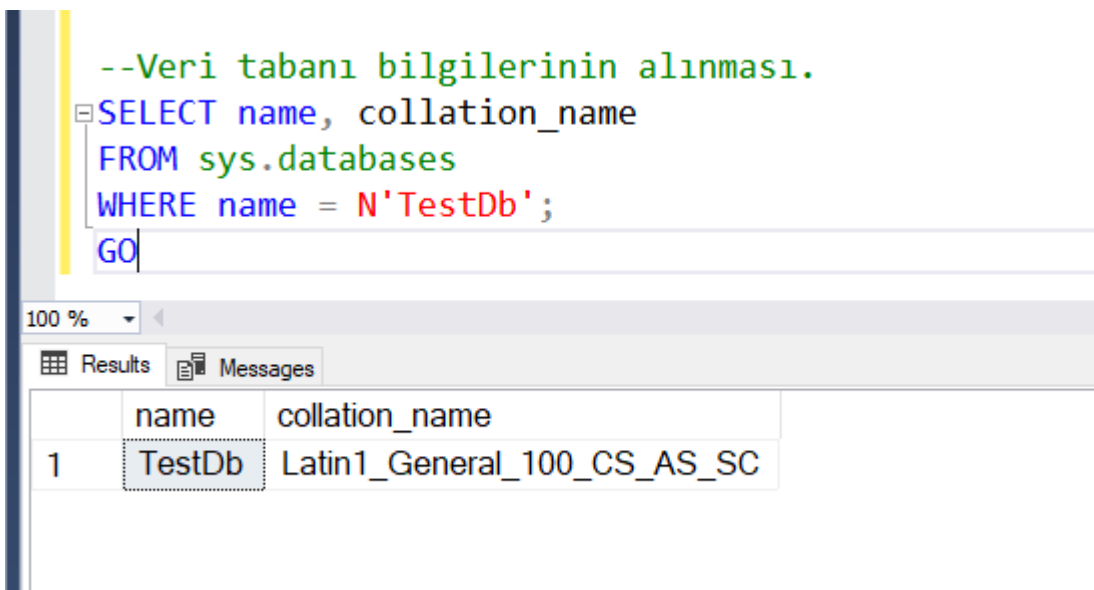
Sunucu seviyesinde collection değiştirilmeden sistem veri tabanlarına ait collection ayararı değiştirilemez.

Veri tabanı collection ayarı, veri tabanına ait bütün

bilgilerde geçerli olur. Bir veri tabanına ait collation ayarı değiştirildiğinde geçici tablolarla çalışan sorgularında hata alınabilmektedir. Alınan bu hatanın sebebi, geçici tabloların tempdb veri tabanında tutulmasında dolayısıdır. Bu tür alınabilecek hataların önüne geçmek için COLLATE komutu ile sorgu düzeyinde collation ayarı yapılmalıdır.

Veri tabanı oluşturulurken collation ayarının belirtilmesi ve oluşturulan veri tabanına ait bilgilerin sorgulanması;

```
USE master;
GO
CREATE DATABASE TestDb
COLLATE Latin1_General_100_CS_AS_SC;
GO
--Veri tabanı bilgilerinin alınması.
SELECT name, collation_name
FROM sys.databases
WHERE name = N'TestDb';
GO
```



The screenshot shows a SQL query window with the following text:

```
--Veri tabanı bilgilerinin alınması.
SELECT name, collation_name
FROM sys.databases
WHERE name = N'TestDb';
GO
```

Below the query window, there is a 'Results' pane showing a table with two columns: 'name' and 'collation\_name'. The table contains one row with the values 'TestDb' and 'Latin1\_General\_100\_CS\_AS\_SC'.

	name	collation_name
1	TestDb	Latin1_General_100_CS_AS_SC

Aşağıdaki örnek kodda olduğu gibi veri tabanına ait collation

ayarı değiştirilebilir. Veri tabanı seviyesinde yapılan collation ayar değişikliği kolon seviyesinde yapılmış olan ayarları etkilememektedir.

```
ALTER DATABASE myDB COLLATE Greek_CS_AI;
```

Aşağıdaki kod ile de belli bir veri tabanına ait collation ayarına ulaşılabilir.

```
SELECT CONVERT (VARCHAR(50),  
DATABASEPROPERTYEX('database_name','collation'));
```

## **Kolon Seviyesinde Collation Ayarı**

Tablo oluşturulurken COLLATE komutu ile her bir kolonun collation ayarı ayrı ayrı olarak ayarlanabilir, ayarlanmaz ise veri tabanına ait collation ayarı varsayılan olarak seçilir.

ALTER TABLE ile mevcutta olan kolonların ayarları değiştirilebilir.

```
ALTER TABLE tabloAd ALTER COLUMN kolonAd NVARCHAR(10) COLLATE  
Greek_CS_AI;
```

## **Sorgu Bazlı Collation Ayarı**

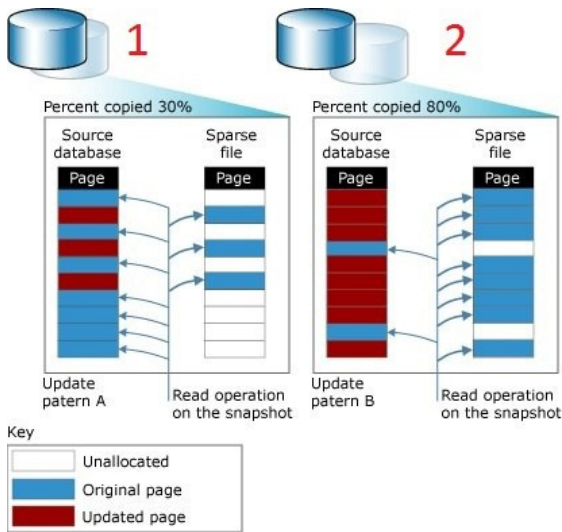
Sorgu bazlı collation ayarında sorgu çalışma zamanında COLLATE komutu ile ayarlanır. Özellikle ORDER BY işlemlerinde kullanılır.



```
SELECT isim FROM urunler ORDER BY name COLLATE Latin1_General_CS_AI;
```

Collation ayarı veri tabanında sorguların doğru sonuç döndürmesi için önemli olduğu kadar performans için de önemli olduğundan daha projelerin kavramsal çalışmalarında karar verilmesi gereken bir konudur.

# SQL Server Veri Tabanı Snapshot Alma

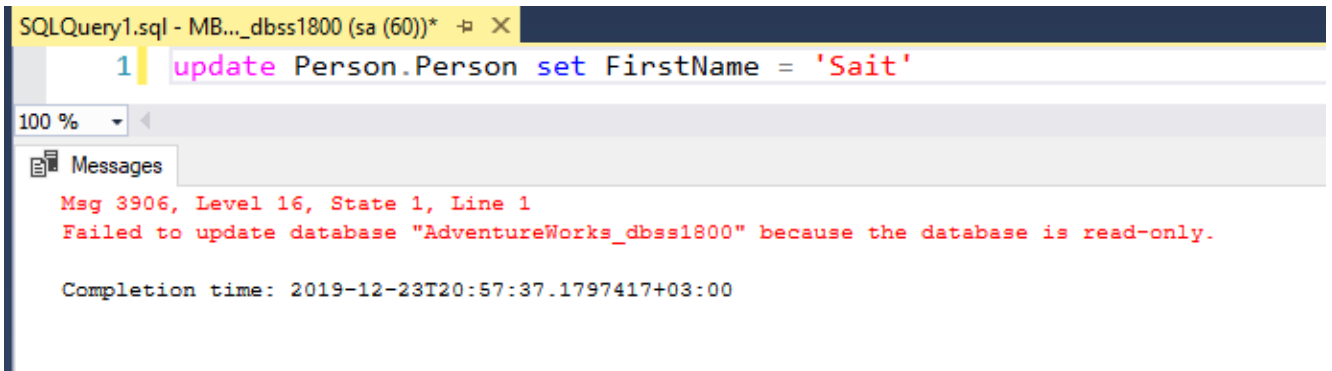


Snapshot ifadesi genel olarak anlık sistemin anlık fotoğrafı olarak tanımlanabilir. Snapshot alma amacı, sistemin anlık görüntüsünü alıp sonradan sistemi önceki haline döndürmek gerekmesi durumunda döndüre bilmektir. Snapshot alındıktan sonra anlık görüntü için bir fark dosyası oluşturulur. Bu andan 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.

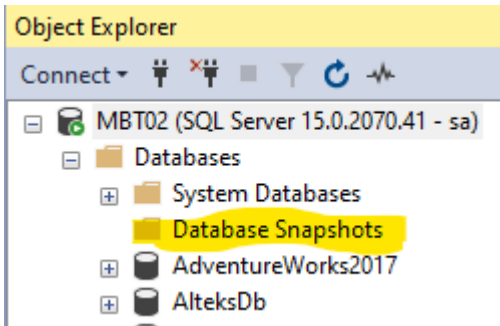
Oluşturulan bu snapshot veri tabanları salt okunur veri tabanlarıdır ve sadece select cümleciğini desteklerler.



```
SQLQuery1.sql - MB..._dbss1800 (sa (60))*  X
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



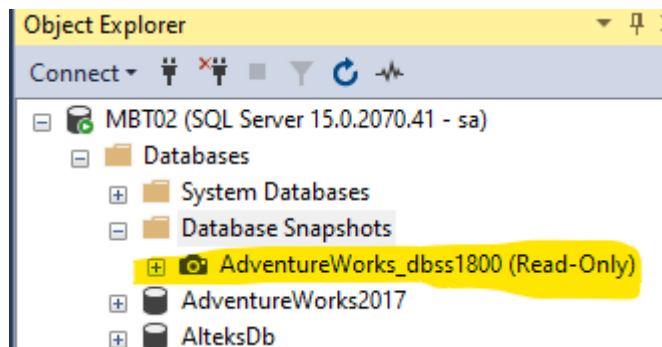
SSMS Database Snapshots

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

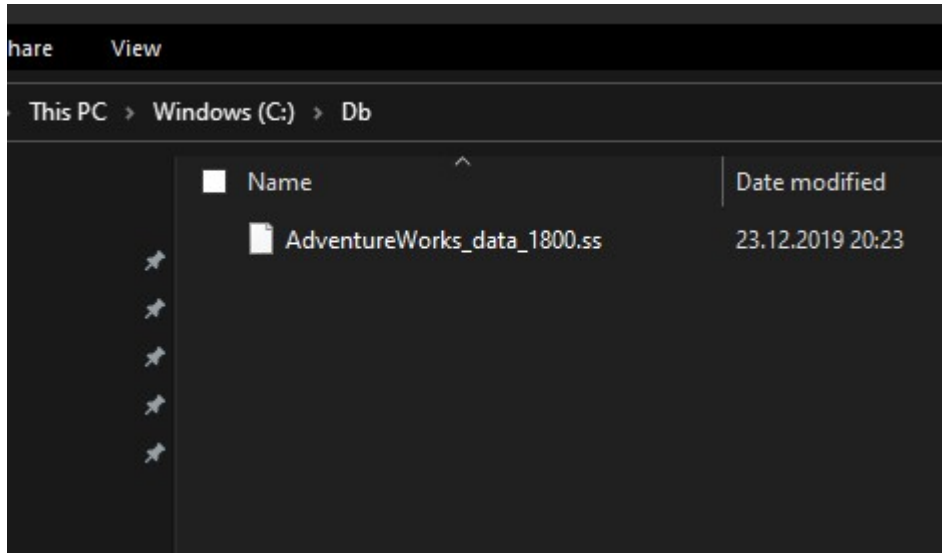
```
CREATE DATABASE snapshot_olarak_olusturulacak_dbAd  
  
ON  
  
(  
  
NAME =Mantıksal_dosya_ad,  
  
FILENAME ='olusturulacak_dosya_tam_ad'  
  
)  
  
AS SNAPSHOT OF kaynak_dbAd
```

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
```

100 %

Results Messages

	(No column name)	FirstName	LastName
1	Orjinal 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.

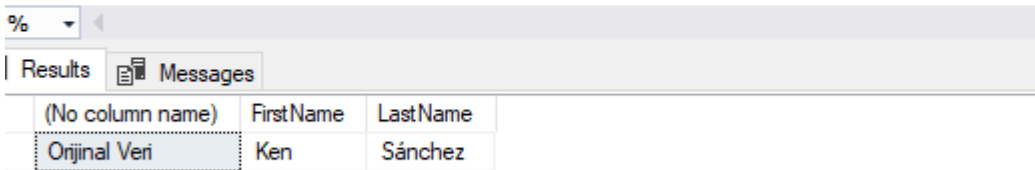
Completion time: 2019-12-23T20:48:26.4463731+03:00
```

```
USE master;
-- AdventureWorks2017 veri tabanını AdventureWorks_dbss1800
zamanına döndürme
```

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

Geri dönüş olduktan sonra az önce deđiřtirdiđimiz 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

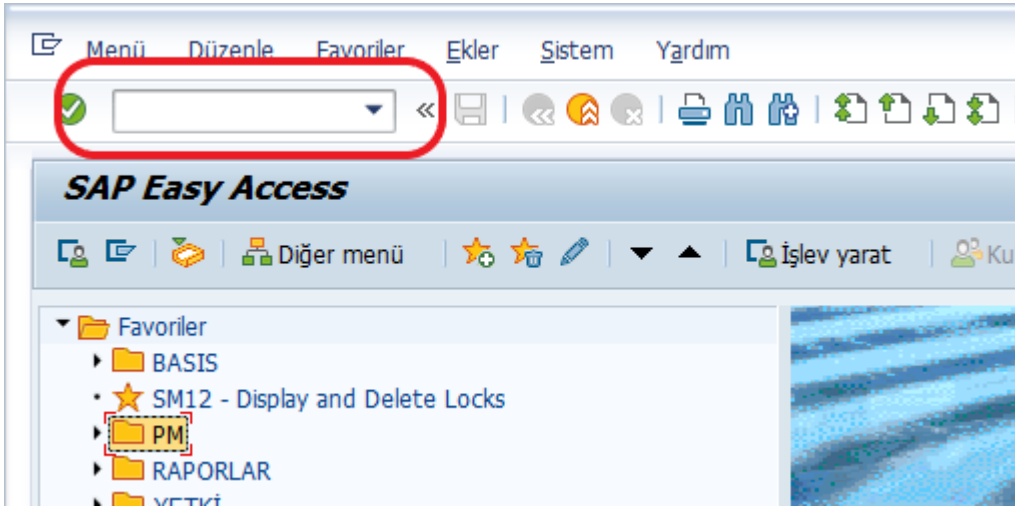
Yazımızı bitirmeden snapshot veri tabanları konusunda söylememiz gereken son bir konu da, snapshotların kendi başlarına bir veri tabanı olmadıkları ve sistemde yaşamaları kaynak veri tabanlarının sistemde yaşıyor olması gerekliliđidir.

---

## SAP T-Code (Transaction Code) Kavramı

SAP sisteminde, Windows uygulamalarını aksine ekranla girmek için aktif bir menü bulunmamaktadır. Bunun yerine her ekrana

atanmış bir kod bulunmaktadır. Ekranlara atanan bu kodlar SAP ekranında sol üst köşede bulunan komut alanına girilir.



### SAP Komut Alanı

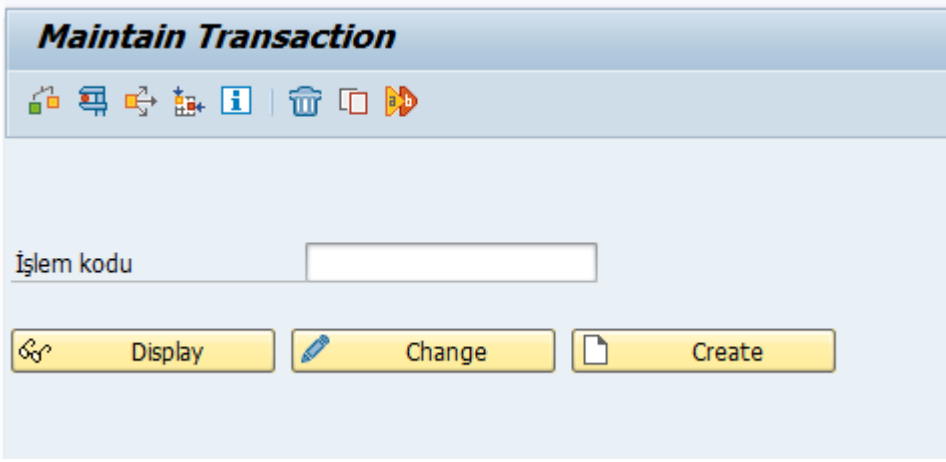
Bir T-Code, harf, rakan veya ikisinden oluşur. SAP'de kendisinin standart ekranlarında olduğu gibi kullanıcı tarafından geliştirilen ekranlara ulaşmak için de T-Code tanımlamak gerekmektedir.

SAP ana ekranda iken komut alanına gidilecek ekranın kodu yazılıp ENTER tuşuna basıldığında veya komut alanınının solunda bulunan yeşil butona tıklandığında ilgili ekran açılacaktır. Ancak herhangi bir ekran içerisinde iken başka ekrana geçilmek istendiğinde gidilecek ekranın t-kodunun başına /o veya /n eklemek gerekmektedir.

/n	İçinde bulunulan ekranı kapatıp aynı pencerede gidilecek ekranı açar. Örnek: /nse38 (se38 gidilmek istenen ekran)
/o	İçinde bulunulan ekranı kapatmadan gidilecek ekranı yeni pencerede açar. Örnek: /ose38 (se38 gidilmek istenen ekran)

# Sistemde Tanımlı T-Code'ların Görüntülenmesi ve Deđiştirilmesi

SAP ierisindeki hemen hemen her yapıyı veri tabanında tutar ve bir ekran üzerinden kontrollerini sađlar. Sistemde bulunan T-kodlara da SE93 Transection Code (İşlem Kodu) ile ulaşılabılır.



SE93 ekranına girildikten sonra mevcutta olan bir işlem kodunu (T-Code) görüntülemek veya deđiştirmek için "İşlem Kodu" alanına ilgili T-Code yazılıp yapılmak istenen işleme tıklanır.

Yukarıdaki ekranda örnek olarak "ZPP001" işlem kodunun ayrıntılarını görmek için "İşlem Kodu" alanına "ZPP001" yazıp "Display" butonuna tıklayalım.



### Display Report Transaction

Transaction code 1 ZPP001

Package 2 ZPP

Transaction text 3 Çekilmiş Tel Teyit

Program 4 ZPP\_PR001

Selection screen 5 1000

Start with variant

Authorization Object

Values

#### Classification

Transaction classification

Professional User Transaction

Easy Web Transaction Hizmet

Pervasive enabled

#### GUI support

SAPGUI for HTML

SAPGUI for Java

SAPGUI for Windows

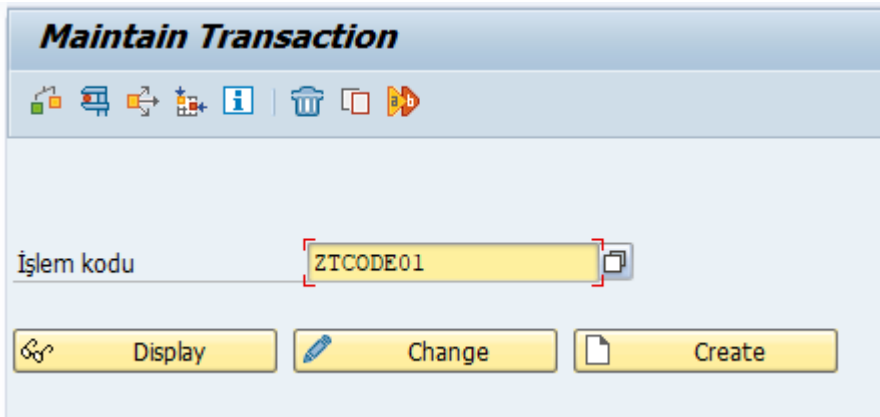
İşlem kodunun detayında yer alan bilgilerin açıklamalarını şöyle yapabiliriz;

1. İşlem kodunun kendisi
2. İşlem kodunun içerisinde çalıştığı paketin ismi
3. İşlem koduna girilirken görünecek ekran başlığı
4. İşlem koduna bağlı olan ABAP programı. İşlem kodu, komut alanına yazılıp ENTER'a basıldığında bu program çalışacak.
5. İşlem kodunun programı çalıştırırken çalıştıracacağı başlangıç ekran numarası

## Yeni T-Code Tanımlama

Sisteme yeni T-Code eklemek için gene SE93 ekranına gidip “İşlem Kodu” alanına tanımlanmak istenen T-Code girilip “Create” butonuna tıklanır.

NOT: SAP sisteminin bir standardı olarak kullanıcı tanımlı nesnelere Y veya Z harfi ile başlamak zorundadır. Örneğin kullanıcı tanımı olarak TCODE01 isimli bir T-Code tanımlanamaz, bunun yerine YTCODE01 veya ZTCODE01 seçeneklerinden biri seçilmelidir.



The screenshot shows the SAP 'Maintain Transaction' (SE93) interface. The title bar is 'Maintain Transaction'. Below the title bar is a toolbar with icons for search, print, refresh, help, delete, and save. The main area has a label 'İşlem kodu' (Transaction Code) and a text input field containing 'ZTCODE01'. Below the input field are three buttons: 'Display' (with a magnifying glass icon), 'Change' (with a pencil icon), and 'Create' (with a document icon).

“Create” butonuna tıklandıktan sonra gelen ekranda;

The image shows a screenshot of the 'Create Transaction' dialog box in SAP. The window title is 'Create Transaction'. The 'Transaction code' field contains 'ZTCODE01'. Below this, there is a section titled 'Transaction attributes' which includes a 'Short text' field with a checkmark. Underneath, there is a 'Start object' section with five radio button options: 'Program and screen (dialog transaction)', 'Program and selection screen (report transaction)', 'Method of a class (OO transaction)', 'Transaction with variant (variant transaction)', and 'Transaction with parameters (parameter transaction)'. The first option is selected. At the bottom right, there are two buttons: a green checkmark and a red 'X'.

“Short Text” alanına oluşturulacak T-Code’a ait kısa açıklama girilir.

“Start Object” alanında da bu işlem kodunun bağlanacağı programın tipi seçilir ve yeşil tik butonuna tıklanır.

### Create Dialog Transaction

Transaction code: ZTCODE01

Package:

Transaction text: Deneme Kod

Program:

Screen number:

Authorization Object:  Values

Maintenance of standard transaction variant allowed

#### Classification

Transaction classification

Professional User Transaction

Easy Web Transaction Hizmet

Pervasive enabled

#### GUI support

SAPGUI for HTML

SAPGUI for Java

SAPGUI for Windows

Son ekranda da T-Code görüntüleme ve deęiřtirme bařlıęında belirtilen bilgilere gre giriřler yapıldıktan sonra ekranın yukarısında bulunan “Kaydet” butonu ile T-Code sisteme kaydedilir. Kaydetme iřlemi sırasında gelen ekranda ilgili paket ve sistemler arasında tařıma iin requestler seilerek iřlem tamamlanır.

Create Object Directory Entry

Object R3TR TRAN ZTCODE01

Attributes

Package

Person Responsible

Original System MSD

Original language TR Türkçe

Created On

Local Object Lock Overview