**12.06.2023 1. Ders**

**Veri**: işlenmemiş bilgidir. Hayatımızdaki her şey.

**Yapısal veriler:** Yapısal formattaki veriler. Verinin belli standard/format varsa. Ad soyad mail …(Sayısal, tarihsel, metinsel veriler)

**Yapısal Olmayan Veriler:** Binary Veriler (Parmak izi,ses, fotoğraf verisi. )

Şirketler neden veriyi kaydeder?

1. **Veri Analizi ve Raporlama İhtiyacı**: Daha sonra veriye bakarak geçmişte ne yapmıştık resmi için.
2. Var olan veriye bakarak **geleceği tahminleme** çalışması yapabilmek için de veriyi kaydetmeleri gerekiyor .

Şirketler verileri nereye kaydediyorlar

* Excele çok kayıt ediliyor.

Veritabanı sistemleri:

* **Relational Database Management System – RDBMS** (İlişkisel veritabanı yönetim sistemi.) Veri tutarlılığını garanti altına alır.
  + Oracle
  + MSSQL Server
  + IBM Db2
  + MySQL
  + PostgreSQL : Lisans ücreti yok.

**12.06.2023 2. Ders**

Tablo: Satır ve sütunları bulunan, verinin saklandığı yerdir.

Logical Design 🡪 Neler lazım onlar belirlenir.

Physical Design 🡪 Mantıksal tasarımda belirlenen bilgileri normalizasyon kurallarıyla oluşturmamız lazım.

**Normalizasyon Kuralları Pozitif Tarafları**

İşlemlerin hızlı çalışabilmesi için normalizasyon kurallarını kullanmak lazımdır.

Eş zamanlılık artar.

**Normalizasyon Kuralları Negatif Tarafları**

Veriyi okumak istediğimizde birden çok tabloya gidip join yazmamız gerekecek. Join arttıkça sorgu süresi artmaktadır.

**Normalizasyon Kuralları**

1. **Aynı kolon içerisinde tekrar eden veri olmasın**
2. **Tekrar edecek veri varsa o verileri tutacak ayrı bir tablo oluştur.**
3. **Tabloları parçaladıktan sonra tabloları ilişkilendirmen lazım.**

**OLTP Database 🡪**

**Identity:** Kolonun otomatik Numara özelliğinin açılması.

**Primary Key**: Unique değer atanmasını sağlar. Aynı değerin ikinci kere girilmesini engeller.

**Foreign Key**: İki tablonun arasında ilişki oluşturmaya yarıyor. (İlişkisel veri tutarlılığı sağlamak. Bir tabloda işlem varken karşılığı var mı diye control sağlanır. Tutarsızlık olmasını engelliyor)

A screenshot of a computer

Description automatically generated with low confidence

**DataWareHouse(DWH)**

Geceden geceye veri ambarına aktarılıyor. Veri ambarından raporu çekin diyolar. Veri ambarları denormalizasyona göre oluşturuluyor. Prod datasını yormamış oluyoruz.

Child kayıtlar varken parent kayıt silebilmek için:

1. Alt tablolara gidip silinecek parent ile ilgili childları silip silinebilir.
2. Cascade ayarı seçilirse foreign key için hata verme, child tabloları otomatik olarak sil.

Şema Adı: Tabloları bir arada tutmaya yarayan konteynırdır..

Tablo Adı:

**12.06.2023 3. Ders**

**HumanResources.Department**

**HumanResources 🡪 Şema , Department 🡪 Tablo**

**. öncesi:** Şema bir arada görmek istediğin tablolar için container görevi görürüm. Bir arada tutmak istediğimiz tabloları bir arada tutmamızı sağlar.

**. sonrası:** Tablo

Şema belirtilmezse otomatik olarak Şema kendiliğinden oluşturulur. ‘dbo’ şeması otomatik oluşturulan şemanın adıdır.

Tablolardan nasıl sorgulama yapılır.

New Query yazısına tıklayarak sorgu ekranı açılır.

Kod yazmadan önce hangi veritabanında çalışacağını doğru seçtiğinize emin olunuz. ‘Available Databases’ dropdown listen seçilir.

Select \* from Tablo Database.Şema.TabloAdı yazılır.

Mouseu getirip \* üstünde tutunca yer alan kolonların adları yer alır.

Sorgu nasıl çalıştırılır.

Sorgu seçilir.

1. Execute butonuna tıklanır.
2. F5 tuşuna basılır.
3. Alt + X tuşlarına aynı anda basılır.

Kodu parça parça bölmek okunabilirlik için önerilir.

Kaç satır geldiğini sağ altta yer alan … rows alanından görülebilir.

Select Kolon1, Kolon2, Kolon3 from Database.Şema.TabloAdı

Otomatik tamamlamanın kolon isimlerini getirebilmesi için tablo adını bilmesi lazım. Select from tablo adı yazılır. Sonra select yapılacak kolon isimleri otomatik tamamlamayla getirilebilinir.

**12.06.2023 4. Ders**

Yorum:

1. -- 🡪 Tek satır yorum yazmak
2. /\*….\*/ 🡪 Araya yazılan tüm değerler yorum olarak yazılır.

Select \* from database.şema.tablo where kolon1 = ‘…’ and kolon2 = 000

Eşit Değildir:

1. !=
2. <>

Arasında

1. Kolon adı between BAŞLANGIÇ DEĞERİ and BİTİŞ DEĞERİ (Between’de sayıları da dahil eder.)
2. Kolon Adı > BAŞLANGIÇ DEĞERİ and kolon adı < BİTİŞ DEĞERİ

Büyük Eşittir / Küçük Eşittir

1. >=
2. <=

AND

OR

**12.06.2023 5. Ders**

IN operatörü

IN(değer1, değer2,….,değerN)

NOT IN operatörü

Null değerleri getirmez.

Null🡪 Boş bırakılmış. (Excel blank)Filtrelemede =, !=, <, > , in kullanılamıyor.

IS NULL

IS NOT NULL

Tarih Filtreleme

1. YEAR(KOLON ADI) = 2012
2. \*\*\* BETWEEN AND ÇOK TERCİH EDİLMEZ TARİHLER İÇİN

Date format: Date between ‘2012-01-01’ and ‘2012-12-31’ => saat bilgisi yazılmazsa otomatik olarak 00:00:00 saatini getirir sadece. Bunu engellemek için between yerine > / < kullanılır.

1. BEST PRACTICE TARİHLER İÇİN

OrderDate >= ‘2012-01-01’ and OrderDate < ‘2013-01-01’

Tarih tek tırnak içerisine yazılır. Yıl-ay-gün kullanılırsa hep doğru çalışır.

**13.06.2023 1. Ders**

LIKE operatörü

% 🡪 0 ya da daha fazla karakteri temsil eder

\_ 🡪 1 adet karakteri temsil eder.

like '%[%]%'

distinct 2 alan ile yazılırsa ikisini birleştirip tekilleştirir.

distinct

birden fazla kolonu tekilleştirmek için: distinct Kolon\_1, Kolon\_2

**13.06.2023 2. Ders**

SQL’de sorgu içinde if kullanamazsın!

Storch prosedür, function, trigger: prosedür kodu yazarken if kullanılabilir.

Select Kolon\_1, Kolon\_2, …, Kolon\_N,

Case when Kolon\_adı kriteri then ‘…’

when Kolon\_adı kriteri then ‘…’

else ‘….’

End as ‘Verilecek İsim’

From Database.Şema.Tablo

CASE WHEN

Dil alfabesi ayarı: collation ayarı. Hangi dil kullanacağı en baştan seçilir.

Unicode karakter olarka göster

Case when ListPrice > 2000 then N’Pahalı’ 🡺 Nasıl yazıdysan öyle yazdırabilirsin.

**13.06.2023 3. Ders**

Order by KOLON ADI asc & desc

Asc & desc belirtilmezse ascending olarak sıralar.

Order by Kolon\_1 asc, Kolon\_2 desc

Select top 10 aaa, bbb from

Select top 10 percent aaa, bbb from

Select top 10 with ties aaa,bbb from

**13.06.2023 4. Ders**

**Pagination 🡪** Offset … rows fetch next .. rows only

select Name, Color, ListPrice

from Production.Product

order by ListPrice desc

offset 20 rows fetch next 20 rows only.

Yazılış Sırası

SELECT ……………(5)

FROM ……………(1)

WHERE ……………(2)

GROUP BY ……………(3)

HAVING ……………(4)

ORDER BY ……………(6)

SELECT NAME AS “ÜRÜN ADI”, LISTPRICE AS [LİSTE FİYATI]

**13.06.2023 4. Ders**

Count fonksiyonu null olanları saymaz

SUM

MIN

MAX

COUNT

AVG

Select KOLON\_1, SUJM(KOLON\_2)

From database.şema.tablo

group by KOLON\_1

-- her bir renkten kaç adet ürün var

select Color,sum(ListPrice)

from Production.Product

group by Color

-- her bir renkten kaç adet ürün var bu sayede hepsi null da dahil gelir

select Color,count(\*) as Adet

from Production.Product

group by Color

4526452

JOINS

**14.06.2023 1. Ders**

1- Inner Join

2- Outer Join

* Left Join
* Right Join
* Full Join

3- Cross Join

A picture containing screenshot, diagram, text, circle

Description automatically generated

A screenshot of a computer

Description automatically generated

N’Black’ Unicode desteği

1. Sub querylerde dışarıdaki sorgunun filtrelenen kolonu ile içerdeki sorgunun selectinden dönen kolon aynı veri tipinde olmalıdır. Sayıysa sayı, metinse metin, tarihse tarih
2. Sub querylerde parantez içerisindeki sorgudan birden fazla satır sonuç geliyorsa dışarıdaki sorgunun filtreleme kısmında = yeriine IN operatorü kullanılmalıdır.

Sclar sonuç 🡪 Tek bir sonuç

A picture containing text, screenshot, diagram

Description automatically generated

SET Operator Kuralları

1. Select içerisinde yer alan kolon sayıları eşit olmalı
2. Kolonların veri tipleri aynı olmalı

**15.06.2023 1. Ders**

Veri Tipi: Kolona girilecek verileri kısıtlamak için kullanılır. Değişken ve parametreler için de veri tipleri seçilir.

1. Numerical Values
2. Date Values
3. String Values
4. Sayısal Veri Tipleri

Tam sayılar

* tinyint 🡪 1 byte -- 255 -- Müşteri Yaşı, Anket Puanı
* smallint 🡪 2 byte -- 65536 --
* Int 🡪 4 byte -- 2 bio 147 mio --
* bigint 🡪 8 byte -- 2^63 --

Ondalıklı sayılar

* decimal 🡪 Standard decimal(p,s) --precision,scale
* numeric 🡪 Microsoft SQL numeric(p,s) -- precision,scale
* floa t 🡪 1 <>1.0 (Scientific) -- DO NOT USE (Microsoft)
* real 🡪 1 <>1.0 (Scientific) -- DO NOT USE (ISO)
* money 🡪 virgülden sonra 4 hane -- DO NOT USE - depreciated
* smallmoney 🡪 virgülden sonra 4 hane -- DO NOT USE - depreciated

decimal(5,2)

134.67 🡪 Tutulur

1223.8🡪 TUTULMAZ

23.67789456 🡪 23.68 olarak tutulur

* bit 🡪 0(false) veya 1(True)

Tarihsel Veri Tipleri

* datetime YYYY-MM-DD HH:MI:SS 1753-9999 yılları arası
* smalldatetime YYYY-MM-DD HH:MI:SS 1900-2079

--------------------------------------------------------------------------------------

* date YYYY-MM-DD 0001-9999
* time HH:MI:SS 00:00:00-23:59:59
* datetime2 YYYY-MM-DD HH:MI:SS 0001-9999
* datetimeoffset YYYY-MM-DD HH:MI:SS+-TZ 0001-9999 (Greenwich saat farkı ekler)

Metinsel Veri Tipleri

* char(max\_value) fixed length – max value kadar giriş yapılmalı -

(PostaKodu,TCKN,IBAN)

Max value 🡪 1 page = 8 kb 🡪8192 byte 🡪 yaklaşık 192 byte(reserved) 🡪 8000

1 karakter 🡪 1 byte

char(10) 🡪 ‘Cagatay ’ 🡪 Cagatay+blank+blank+blank

* nchar(max\_value) fixed length – max value kadar giriş yapılmalı

Max value 🡪 1 page = 8 kb 🡪8192 byte 🡪 yaklaşık 192 byte(reserved) 🡪 4000

1 karakter 🡪 2 byte

nchar(10)🡪 ‘Alp ‘ 🡪 Alp +blank+blank+blank+blank+blank+blank+blank

* varchar(max\_value) not fixed length –

Max value 🡪 1 page = 8 kb 🡪8192 byte 🡪 yaklaşık 192 byte(reserved) 🡪 8000

1 karakter 🡪 1 byte

nchar(10)🡪 ‘Alp‘

* nvarchar(max\_value) not fixed length – Unicode desteği var

Max value 🡪 1 page = 8 kb 🡪8192 byte 🡪 yaklaşık 192 byte(reserved) 🡪 4000

1 karakter 🡪 2 byte

nchar(10)🡪 ‘Cagatay‘

----------------------------------------------------------------------------------

8000 karakter limitini geçmek için aşağıdakiler kullanılır.

* Text DO NOT USE - depreciated
* ntext DO NOT USE - depreciated
* varchar(max)

2 GB 🡪 2 milyar karakter

* nvarchar(max)

2 GB 🡪 2 milyar karakter

Uzunluğu sabit mi olacak

Unicode desteği olacak mı?

Veri tipi Dönüşümleri

Xml, varbinary (ses, foto video binary verileri tutmak için), uniqueidentifier (GUID), geometry(ccordinates- x and y), geography (ccordinates, enlem boylam olarak)

Data Type Conversion

CAST() – ASCII standard

CAST(kolonadi as veritipi)

CONVERT() – Microsoft Standard

CONVERT(veritipi,kolonadi)

Database üzerine sağ tıklayıp Create New Table yazısına tıklanır.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Identity 🡪 kolonun otomatik Numara üretme özelliğidir.

A screenshot of a computer

Description automatically generated

Sarı Anahtar 🡪 PrimaryKey

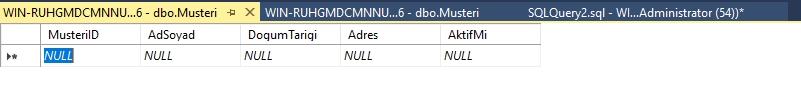
Schema oluşturmak için:

Alter table transfer scheme

Veri Ekleme / Güncelleme / Silme

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

Sanal tablolar ve geçici tablolar

16.06.2023 4. Ders

Sıra numarası üretme

1. RANK() --11111666610101010

RANK() OVER(order by Kolon\_Adı asc/desc)

1. DENSE\_RANK()--11111122223333

DENSE\_RANK() OVER(order by listprice desc)

1. ROW\_NUMBER()--123456789

ROW\_NUMBER() OVER(order by listprice desc) as Sirano

Değer Yuvarlama

1. ROUND(KolonAdı, 2(virgülden sonraki hane sayısı)
2. FLOOR(KolonAdı) 🡪 Aşağı yuvarlar
3. CEILING(KolonAdı) 🡪 Yukarı yuvarlar

LEFT(KolonAdı,gelecek karakter sayısı)

RIGHT(KolonAdı,gelecek karakter sayısı)

SUBSTRING(KolonAdı,başlanacak karakter indexi, bitecek karakter indexi)

REPLACE(KolonAdı, ‘Değiştirilecek’, ‘Eklenecek’)

DATEADD(YEAR,2,KolonAdı)

DATENAME(WEEKDAY,Tarih Kolonu)

DATEDIFF(MONTH,OrderDate, GETDATE())

select DATEDIFF(HOUR,'1994-02-25',GETDATE())

select DATEDIFF(DAY,'1994-05-08',GETDATE())

https://www.egitimtakvimi.com/Egitim/BeginSurvey?type=Private&surveyNumber=6aedb1

/\*\*\*\*\*\* Script for SelectTopNRows command from SSMS \*\*\*\*\*\*/

SELECT TOP (1000) [optname]

,[value]

,[major\_version]

,[minor\_version]

,[revision]

,[install\_failures]

FROM [master].[dbo].[MSreplication\_options]

select Name, Color, ListPrice from Production.Product

--fiyatı 0'dan farklı olan ürünler

select Name, Color, ListPrice from Production.Product

where ListPrice <> 0

--fiyatı 1000 ile 2000 arasında olan ürünler

select Name, Color, ListPrice from Production.Product

where ListPrice between 1000 and 2000

--fiyatı 1000 ile 2000 arasında olan ürünler

select Name, Color, ListPrice from Production.Product

where ListPrice >= 1000 and ListPrice <= 2000

--rengi siyah olan ürünler

select Name, Color, ListPrice from Production.Product

where Color = 'Black'

--rengi siyah ya da kırmızı olup aynı zamanda fiyatı 1000den büyük olan ürünler

select Name, Color, ListPrice from Production.Product

where (Color='Red' or Color='Black') and ListPrice > 1000

--rengi siyah ya da kırmızı ya da mavi ya da sarı olan ürünler

select Name, Color, ListPrice from Production.Product

where Color IN ('Red','Black','Blue','Yellow')

-- 2012 yılında verilen siparişler gelsin

select \*

from Sales.SalesOrderHeader where YEAR(OrderDate) = 2012

-- Soyadı K ile başlayan kişiler gelsin

select \* from Person.Person where LastName like 'k%'

-- Soyadı K ile biten kişiler gelsin

select \* from Person.Person where LastName like '%K'

-- Soyadının içinde k geçen kişiler

select \* from Person.Person where LastName like '%K\_\_'

--XXX-XXX-XXXX formatına uygun telefon numaralarını getir.

select \* from Person.PersonPhone where PhoneNumber like '\_\_\_-\_\_\_-\_\_\_\_'

--XXX-XXX-XXXX formatına uygun rakam içerecek şekilde telefon numaralarını getir.

select \* from Person.PersonPhone where PhoneNumber like '[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'

-- içinde % karakteri geçen sonuçlar

select \* from Person.PersonPhone where PhoneNumber like '%[%]%'

-- kaç farklı renk mevcut

select distinct Color from Production.Product

--renk ve bedenler birlikte tekilleştirildi

select distinct Color,Size from Production.Product

-- fiyatı 0-1000 --> Ucuz

-- fitayı 1000-2000 --> Orta

-- fiyatı 2000'den büyük --> Pahalı

select Name, Color, ListPrice,

case when ListPrice between 0 and 1000 then 'Ucuz'

when ListPrice between 1000 and 2000 then 'Orta'

when ListPrice > 2000 then 'Pahalı'

else 'Diğer' end as 'Kategori'

from Production.Product

-- Fiyati en pahali olan ilk yuzde 10 luk urun dilimi

select top 10 percent Name, Color, ListPrice

from Production.Product

order by ListPrice desc

select top 10 with ties Name, Color, ListPrice

from Production.Product

order by ListPrice desc

-- fiyatı en pahalı olan 2. 20 adet ürün (21-40 arası) gelsin

select Name, Color, ListPrice

from Production.Product

order by ListPrice desc

offset 20 rows fetch next 20 rows only.

-- her bir renkten kaç adet ürün var

select Color,sum(ListPrice)

from Production.Product

group by Color

-- her bir şehirde kaç adet adres var

select City, COUNT(\*) as Adet

from Person.Address

group by City order by Adet DESC

-- her bir rengin en pahalı fiyatı, en ucuz fiyatı, ortalama fiyatı

select Color,max(ListPrice) as En\_Pahalı\_Fiyatı, min(ListPrice) as En\_Ucuz\_Fiyatı, AVG(ListPrice) as Ortalama\_Fiyatı

from Production.Product

group by Color

-- her bir yılda toplam ne kadar ciro elde edildi

select Year(OrderDate), SUM(SubTotal)

from Sales.SalesOrderHeader group by Year(OrderDate)

order by YEAR(OrderDate)

select \*

from Sales.SalesOrderHeader

-- her bir yilin her bir ayinda toplam ne kadar ciro elde edildi

select Year(OrderDate) as Yıl, Month(OrderDate) as Ay, SUM(SubTotal) as Ciro

from Sales.SalesOrderHeader group by Year(OrderDate), Month(OrderDate)

order by YEAR(OrderDate), Month(OrderDate)

-- Sipariş tarihi, tutar, bölgeadı gelsin

select Sales.SalesOrderHeader.OrderDate, Sales.SalesOrderHeader.SubTotal, Sales.SalesOrderHeader.TerritoryID, Sales.SalesTerritory.Name from Sales.SalesOrderHeader

left join Sales.SalesTerritory

on Sales.SalesOrderHeader.TerritoryID = Sales.SalesTerritory.TerritoryID

select \* from Sales.SalesTerritory

-- TerritoryID kolonu hangi tablolarda mevcut

select \*

from sys.tables

select \*

from sys.columns

-- Territory ID hangi tablolarda yer almaktadır?

select t.name as TabloAdi, c.name as KolonAdi

from sys.tables as t inner join sys.columns as c

on t.object\_id = c.object\_id

where c.name like '%TerritoryID%'

-- Sipariş tarihi, tutar, bölgeadı gelsin

select t1.OrderDate, t1.SubTotal, t1.TerritoryID, t2.Name from Sales.SalesOrderHeader as t1

left join Sales.SalesTerritory as t2

on t1.TerritoryID = t2.TerritoryID

-- Ürün adı, Renk, Fiyatı, Modeladı gelsin

select \* from Production.Product as p

select \* from Production.ProductModel as pm

select p.Name, p.Color, p.ListPrice, pm.Name from Production.Product as p

inner join Production.ProductModel as pm

on p.ProductModelID = pm.ProductModelID

-- ÜRünlerin kategori adı, alt kategori adı, ürün adı, rengi, fiyatı, bağlı olduğu model adı

select \* from Production.ProductCategory as pc

select \* from Production.ProductSubcategory as psc

select \* from Production.Product as p

select \* from Production.ProductModel as pm

select pc.Name, psc.Name, p.Name, p.Color, p.ListPrice, pm.Name

from Production.ProductCategory as pc

inner join Production.ProductSubcategory as psc

on pc.ProductCategoryID = psc.ProductCategoryID

inner join Production.Product as p

on psc.ProductSubcategoryID = p.ProductSubcategoryID

inner join Production.ProductModel as pm

on pm.ProductModelID = p.ProductModelID

--en son verilen siparişle aynı günde verilen bütün siparişler gelsin

select \* from Sales.SalesOrderHeader where OrderDate IN (

select top 1 OrderDate

from Sales.SalesOrderHeader order by OrderDate desc )

-- fiyatı ortalama fiyattan daha büyük olan ürünler gelsin

select \* from Production.Product where ListPrice >(

select AVG(ListPrice) from Production.Product )

-- en pahalı ürünle aynı renge sahip olan bütün ürünler gelsin

select \* from Production.Product where Color = (

select top 1 Color from Production.Product where ListPrice = (

select MAX(ListPrice) from Production.Product))

select \*

from Person.PersonPhone

where PhoneNumber like '%-%-%'

EXCEPT

select \*

from Person.PersonPhone

where PhoneNumber like '\_\_\_-\_\_\_-\_\_\_\_'

--kişinin adını ve soyadını birleştirilip tek kolonda adsoyad olarak gelsin

select FirstName, LastName, CONCAT(FirstNAme,LastName) as İsim\_Soyisim

from Person.Person

--kişinin adını ve soyadını birleştirilip tek kolonda adsoyad olarak gelsin

select FirstName, LastName, FirstNAme + ' ' + LastName as İsim\_Soyisim

from Person.Person

/\*

Tablonun yanındaki +'ya tıkla

Columns'a tıkla

Açılan ekranda column'ın veri tipini görebilirsiniz

\*/

-- kişilerin sicilno ile soyadı birleştirilip tek kolonda getirilsin

select BusinessEntityID, LastName, CONVERT(nvarchar,BusinessEntityID) + LastName as Sicil\_İsim

from Person.Person

-- kişilerin sicilno ile soyadı birleştirilip tek kolonda getirilsin

select BusinessEntityID, LastName, CAST(BusinessEntityID as nvarchar) + LastName as Sicil\_İsim

from Person.Person

-- sipariş tarihinin tarih kısmı (YYYY-MM-DD) olarak gelsin

select cast(OrderDate as date)

from Sales.SalesOrderHeader

--tarihin formatını değiştirmek için önce string'e döndürmeniz gerekmektedir.

-- style number'a ihtiyacımız var. CONVERT'ü seçip F1'e basınca detaylı şekilde anlatım görebiliriz.

select OrderDate, Convert(nvarchar,OrderDate,104),

Convert(nvarchar,OrderDate,105),

Convert(nvarchar,OrderDate,112),

Convert(nvarchar,OrderDate,130),

Convert(nvarchar,OrderDate,131)

from Sales.SalesOrderHeader

-- tablo olusturma

--şema ismini de belirtebiliriz burada Person.Siparis gibi yapmak. Aksi durumda dbo.Tablo adı olarak tablo oluşur.

create table Siparis

(

--ColumnName DataType Nullable? PK? identity?identity(starts from, increment)

SiparisID bigint not null primary key identity,

SiparisTarihi datetime2 not null,

Tutar decimal(15,2),

MusteriID int not null foreign key references dbo.Musteri(MusteriID)

)

select \* from dbo.Musteri

select \* from dbo.Siparis

--veri ekleme

insert into dbo.Siparis(SiparisTarihi, Tutar, MusteriID)

values(GETDATE(), 643.75, 3),

(GETDATE(), 7548.99, 2),

(GETDATE(), 6859.47, 2)

--veri güncelleme

update dbo.Siparis

set Tutar = Tutar\*1.1

where SiparisID = 1

update dbo.Siparis

set Tutar = Tutar\*1.1

where Tutar < 1000

--veri silme

delete from dbo.Siparis

where SiparisID = 3

--explicit transaction TRANSACTION diye uzun uzun da yazabiliriz.

BEGIN TRAN

update dbo.Siparis

set Tutar = 0

where SiparisID = 2

--Onaylar

COMMIT

--İptal Etmek

ROLLBACK

--NOLOCK kullanımı tutarsız olabilir. Tutarlılık garanti altında olmuyor.

--sıra numarası üretme

select Name, Color, ListPrice, RANK() OVER(order by ListPrice desc) as SiraNo,

DENSE\_RANK() OVER(order by listprice desc) as Sirano2,

ROW\_NUMBER() OVER(order by listprice desc, Name asc) as Sirano3

from Production.Product

-- Her rengin en pahalı ilk 3 farklı fiyata sahip ürünleri gelsin

select \* from (select Name, Color, ListPrice,

RANK() OVER(partition by Color order by ListPrice desc) as SiraNo,

DENSE\_RANK() OVER(partition by Color order by listprice desc) as Sirano2,

ROW\_NUMBER() OVER(partition by Color order by listprice desc) as Sirano3

from Production.Product) as tbl

select Round(SubTotal)

from Sales.SalesOrderHeader

--kaç gündür hayattayız

select DATEDIFF(HOUR,'1994-02-25',GETDATE())

select DATEDIFF(DAY,'1994-05-08',GETDATE())

select DATEDIFF(DAY,'2014-02-27',GETDATE())