--Tabloları, veri tabanları veya sema tanımlamak ıcın kullanılır.

CREATE DATABASE Veli

DROP DATABASE Veli

CREATE DATABASE OkulDb1

CREATE DATABASE HepsiOradaDb

on primary

(

Name='HepsiOrada',

FileName='C:\TEST\Deneme.mdf',

Size=10MB,

FileGrowth=5MB,

MaxSize=UNLIMITED

--deriz yada sınırsız ıcın hıc maxsıze falan yazmamıza gerek yok.

)

log on

(

Name='HepsiOrada\_log',

FileName='C:\TEST\Deneme.ldf',

Size=2,

FileGrowth=5%,

MaxSize=100

)

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

USE HepsiOradaDb

CREATE TABLE Kategori

(

KategoriId int primary key identity(1,1),

KategoriAdi varchar(50),

Aciklama varchar(500)

)

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

CREATE TABLE Urun

(

UrunId int primary key identity(1,1),

UrunAdi varchar(50) not null,

StokMiktari int null,

BirimFiyati money,

KategoriId int foreign key references Kategori(KategoriId)

)

-- sonradan sutun eklemeye calısıyorsak soyle olmalı:

ALTER TABLE Urun ADD ResimYolu varchar(100)

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

--resim dosyasını hemen basabılmek ıcın:

ALTER TABLE Urun ALTER COLUMN ResimYolu IMAGE

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

ALTER TABLE Urun DROP COLUMN ResimYolu

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

--tabloyu ucurmak ıcın :

DROP TABLE Urun

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

DROP DATABASE HepsiOradaDb

use master

--o database ı kullanıyo gorundugu ıcın master ı gecıs yaptık ondan sonra drop database dedık

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

--veri tabanından script almak icin. veri tabanı uzaerınde sagtus(tasks)/generate scripts secılır.

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

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

--CONSTRAINT----------

--Baskı olusturan kodlar.baskı olusturan kodlar

--null/not null :sutunun bos bırakılıp bırakılmayacagını kontrol eder.

--identity :kullanıcıya ızın vermeden otomatık artıs saglar

--primary key :benzersızlık saglar.

--foreign key :Girilen deger referans edılmıs dıger tabloda bulunmasını kontrol eder.

--unique :benzersizlik saglıyor,fakat bir tabloda birden fazlakez kullanılabılıyor.(primary key gorevi goruyor aslında fakat primary key bır tabloda bırtane kullanılır.eger baska tabloya referans etcek ısek primary key ama o sekılde degılse unıque kullanılır.)

--default :sutuna varsayılan bır deger gırısı yapar.

--check :sutundaki belli bir kriteri test ederek veri girisine izin verir.

use Northwind

CREATE TABLE Kitap

(

KitapId int identity(1,1) primary key,

KitapAdi varchar(50) not null,

KategoriId int foreign key references Categories(CategoryId),

ISBN\_No varchar(50) unique,

YazarAdSoyad varchar(100) default 'John Doe',

BasimYili date check (year(BasimYili)>2000)

)

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

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

--VIEW:göruntu ,manzara

--belırlı bır sorguyu sıstem uzerıne kaydetmemızı saglayan yapıdır.

--aynı sorguların tekrar tekrar yazılmasını yada cok uzun ve karmasık sorguların daha rahat yonetılmesını saglar.

--syntaxı:

--create view isim(donen sutun ısımlerı )

--ayarkodu

--as

--select sorgusu

--ayar kodu

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

CREATE VIEW Urunler

as

SELECT p.\*,c.CategoryName FROM Products p JOIN Categories c on p.CategoryID=c.CategoryID

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

--tablolara ne yaptırılabılıyorsa vıew lere de yaptırılabılır

SELECT \* FROM Urunler

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

--stokta bulunmayan urunlerı vıew ıle goruntuleyınız

CREATE VIEW StoktaOlmayanUrunler

as

SELECT \* FROM Products p WHERE UnitsInStock=0

select \* from StoktaOlmayanUrunler

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

--kullanıcıların gorebılecegı tabloları vıew halıne getırıryoruz.bu vıew tablolar daha hızlı cekılır.

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

--Domates urununu stoktan olmayan urun olarak ekleyınız

--stokta olmayan urunlere 0 dan fazla ıslem yapmamasına nasıl ızın verırız:

ALTER VIEW StoktaOlmayanUrunler

as

SELECT \* FROM Products p WHERE UnitsInStock=0

with check option -- bu sekılde engellerız

INSERT INTO StoktaOlmayanUrunler(ProductName,UnitsInStock) VALUES ('Patates',50)

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

ALTER TABLE Employees ADD SilindiMi bit

--employees tablosuna sutun ekledık

select \* from Employees

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

--silinmeyen personelin adını ve soyadını goruntuleyebılecegınız bıre vıew yazın

ALTER VIEW SilinmeyenPersonelTablosu

with schemabinding --view icinde kodların sılınmesını engellıyor

as

SELECT FirstName,LastName FROM dbo.Employees e WHERE SilindiMi=0

select \* from SilinmeyenPersonelTablosu

--eger sılmeye kalkarsak sılemeyız cunku sılınmesını engelledık

alter table Employees drop column SilindiMi--silemez!!

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

--with schemabinding: bir view icerisinde kullanılan bır kolonun sılınmemesı ıcın bu kod kullanılır.

--with schemabinding yazıldıysa kullanılan kolonların adları mutlaka belırtılmelıdır.

--\* ıle tum kolonlar cekılemez.

--tablonun basına mutlaka dbo. eklenmelidir.

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

--with check option: bir view icerisinde kullanılan where kosuluna uymayan verılerle ıslem yapılmasını engeller.

--where kosuluna uymayan verılerle ıslem yapılacaksa mutlaka dırek tabloya ıslem yapılması gerekır.view e ıslem yapılamaz.

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

--herbir naklıyecı kac tane sıparıs tasımıslar ve bu sıparıslerden kac para kazanmıslar?

CREATE VIEW NakliyeciRaporu(SirketAdi,Kazanc,[Siparis Sayisi])

as

SELECT s.CompanyName,SUM(Freight),COUNT(\*) FROM Shippers s JOIN Orders o on s.ShipperID=o.ShipVia group by s.CompanyName

-------

SELECT \* FROM NakliyeciRaporu

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

INSERT INTO NakliyeciRaporu([Siparis Sayisi],Kazanc,[Siparis Sayisi]) values ('Aras Kargo',123,321)-- hata verır cunku ınsert yapmamızı aggregate fonklar ve joın ler bunu engeller.

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

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

DECLARE @isim varchar(50)='Bilge Adam'

SELECT @isim = FirstName+' '+LastName FROM Employees

SELECT @isim

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

DECLARE @beveragesToplam int

SELECT @beveragesToplam= SUM(UnitsInStock) FROM Products WHERE CategoryID=1

--sum fonk sonucunu tanımladıgımızın ıcıne atabılıyoruz.

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

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

--Bir kritere bakarak bazı islemleri yapan yada bazı islemlerin yapılmasını engelleyen yapılara karar yapıları denır.

--syntax:

--if(kriter)

--begin

--kriter dogruysa yapılması gereken ıslemler:

--end

--else if(kriter)

--begin

--kriter dogruysa yapılması gereken ıslemler:

--end

--else

--begin

--kriter dogru degılse yapılması gereken ıslemler:

--end

go

DECLARE @beveragesToplam int

SELECT @beveragesToplam= SUM(UnitsInStock) FROM Products WHERE CategoryID=1

if(@beveragesToplam>100)

begin

print 'stokta yeterınce mevcut'

end

else

begin

SELECT \* FROM Products WHERE CategoryID=1

end

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

--urunlerımden chai isimli urunun stogumda 100 den az urun varsa mevcut stogu 100 arttırın eger 100 den fazlaysa yeterince urun var mesajı gosterınız

DECLARE @urunStok int

SELECT @urunStok=UnitsInStock FROM Products WHERE ProductName='chai'

if(@urunStok<100)

begin

update Products set UnitsInStock=UnitsInStock+100 WHERE ProductName='chai'

end

else

begin

print 'yeterince urun var'

end

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

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

--exists(select sorgusu): eger select sorgusundan herhangı bir sonus donuyorsa true deger,donmuyorsa false degerı alınır.

--karar yapılarından if true degeri gordugu zaman calısır.

if exists(SELECT \* FROM Products WHERE ProductName='Domates')

begin

print 'urun var'

end

else

begin

print 'Urun yok'

end

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

--sekerlemeler kategorısı eger varsa, kategorı daha once eklenmıs uyarısı verınız eger yoksa ekleyınız

if exists(SELECT \* FROM Categories WHERE CategoryName='Candies')

begin

print 'Kategori daha once eklenmis'

end

else

begin

insert into Categories (CategoryName) values('Candies')

end

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

----STORED PROCEDURE----------------

--Sakli yordamlar anlamındadır.

--Herhangi bir sorguyu bir isim altında toplamaya yarar.

--insert,update, delete ve select sorguları ıcın calısabılırler fakat view sadece select sorgularında calısır.(farkı bu)

--store procedure :dısarıdan parametre alabılıyorlar(select sorguları ıcın vıew den farklı olarak)

ALTER VİEW StoktaOlmayanUrunler

as

SELECT \* FROM Urunler WHERE UnitsInStock=0

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

alter procedure UrunVerStokIle

@miktar int

as

SELECT \* FROM Urunler WHERE UnitsInStock < @miktar

--kullanım seklı:

exec UrunVerStokIle 50

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

ALTER procedure UrunVerStokAraligiIle

@miktarBaslangic int,

@miktarBitis int

as

SELECT \* FROM Urunler WHERE UnitsInStock BETWEEN @miktarBaslangic AND @miktarBitis

EXEC UrunVerStokAraligiIle 30,40

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

--birim fiyatlarini dısarıdan yollayabılcegım,gonderdıgım fıyatların arasındakı urunlerı gosteren bir sp yazınız(sp:stored procedure)

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

--en fazla satılan 6 urunu gosteren sp yazın

-- en son satılan 6 urunumu gosteren sp yazın

--id sini verdıgım musterımın bana kac para kazandırdıgını gosteren sp yazın

--baslangıc ve bıtıs satıs aralıgını verdıgım musterımın sıparıslerını goruntuleyen sp yazın

--dısarıdan verdıgım tarıh arasında her bir calısanımın bana kac para kazandırdıgını gosteren bir sp yazın

--dısarıdan isim ve soyısım verdıgım calısanımın sıparıslerını hangı naklıyecı tasıdıgını gosteren sp yazın

--arastırma konusu: dısarıdan verdıgım ısımde verı tabanı acan sp yazınız.