create database deneme

use deneme

create table ogrenciler(

ogr\_no int,

adi varchar(25),

soyadi varchar(25))

create table ogrencileryeni(

kayit\_no int not null ,

ogr\_no int not null identity(20201000,1),

adi varchar(25),

soyadi varchar(25))

bir tabloda 2 tane identity olmaz.

create table degisken\_deneme(

kayit\_no int not null identity(1,1) primary key,

adi varchar(15),

soyadi varchar(15),

Puani int not null

)

select \* from degisken\_deneme

insert degisken\_deneme values('harun','bakirci',100)

insert degisken\_deneme values('ecrin','bakirci',90)

primary key identity(1,1) ile kullanılabilir.

**Kolon ismi değiştirme:**

Sp\_rename ‘tabloadı.kolon adı’,’verilecek kolon ismi’,’kolon olarak işlem yaptığımı belirityorum’

sp\_rename 'YeniTablo.sıra','sıra\_no','column'

**Tablo ismi değiştirme:**

Sp\_rename‘eski tablo adı’,’değiştirmek istediğim tablo adı’

sp\_rename 'yenitablo','VeriTablom'

**Constraintler kısıtlayıcılardır.**

constraint tipleri:

**default constraint** belirtilen kolonun boş geçilmesi durumunda girilecek olan değer.

Constraintlerin genel yapısı:

Add constraint [constraint adı] Default ‘varsayılan değer’ for [kolon adı]

select \* from YeniTablo

alter table YeniTablo alter column maas int

insert into YeniTablo(adi,soyadi) values('recep','bakirci')

alter table YeniTablo add constraint maasConstraint default 0 for maas

--burada 0 dedim çünkü maas kolonum int eger yazı yazarsam bu constrainti silmek zorunda kalırım

--eger constraint atayacağım kolon varchar olsaydı default olarak belirtilmemiş yazabilirdim.

insert into YeniTablo(adi,soyadi) values('cem','denktaş')

**primary key constraint** ekleme :

alter table YeniTablo add constraint PrimarySıra\_no primary key (sıra\_no)

tabloların altında constraints seçeneğinde o tabloda bulunan constraintleri görebilir. Delete ederek silebilirsin.

Default tan önce veri türünü gir.

create table ogrenci(

kayit\_no int unique,

ogr\_no int identity(20201001,1) primary key,

adi varchar(25),

soyadi varchar(25),

dogum\_yeri varchar(25) default 'Belirtilmemiş.')

create table ogrenci(

kayit\_no int unique,

ogr\_no int identity(20201001,1) primary key,

adi varchar(25),

soyadi varchar(25),

dogum\_yeri varchar(25) default 'Belirtilmemiş.')

select \* from ogrenci

create table ogrenci100(

kayit\_no int ,

ogr\_no int identity(20201001,1) ,

adi varchar(25),

soyadi varchar(25),

dogum\_yeri varchar(25) default 'Belirtilmemiş.'

constraint BA primary key(kayit\_no,ogr\_no))

create table notlar(

ogr\_no int identity(20201001,1) ,

ders\_kodu int,

vize int check (vize>0 and vize<=100),

final int check(final>0 and final<=100),

constraint BA1 primary key(ogr\_no,ders\_kodu))

select \* from notlar

select \* from ogrenci100

--ogrenci100 tablosunda dogum yeri default olarak belirtilmemiştir.

--veri girişi yaparken bu şekilde girersen belirtilmemiş yazar ama veri girişi yaptığın

--sutunları belirt.

insert into ogrenci100(kayit\_no,adi,soyadi) values(4,'mert','sis')

create table urunler(

urun\_kod varchar(4) not null,

urun\_ad varchar(25) not null,

constraint chk check (urun\_kod IN ('A089','A090 ','A091')))

create table urunlerYeni(

urun\_kod varchar(4) not null,

urun\_ad varchar(25) not null,

constraint chkyeni check (urun\_kod like ('A9[0-9][0-9]')))

create table urunlerdeneme1(

urun\_kod varchar(6) not null,

urun\_ad varchar(25) not null,

constraint chkdeneme check (urun\_kod like ('A9 [0-9] [0-9]')))

create table OGR(

ogr\_no varchar(8) not null,

ogr\_ad varchar(25) not null,

constraint chkogr check(ogr\_no like ('2020[0-9][0-9]') or ogr\_no in('19990000','19990001','19990002')))

drop table notlar

yeni ilişki oluşturma:

use deneme

create table ogrenci(

ogr\_no int primary key,

adi varchar(25),

soyadi varchar(25),

bolum varchar(3))

create table ders(

ders\_kodu varchar(5) primary key,

ders\_adi varchar(25))

create table notlar(

ogr\_no int,

ders\_kodu varchar(5),

vize int,

final int,

constraint fk foreign key (ogr\_no) references ogrenci(ogr\_no),

constraint fk1 foreign key (ders\_kodu) references ders(ders\_kodu))

on update cascade on delete cascade kullanmazsam veriyi silemem ama bunu yazarsam

örneğin bir öğrenciyi silersem ya da bir dersi silersem o ders ve öğrenciyle alakalı bütün verileri siler

yazmazsam silmeye izin vermez.

create table notlar(

ogr\_no int,

ders\_kodu varchar(5),

vize int,

final int,

constraint fk foreign key (ogr\_no) references ogrenci(ogr\_no) on update cascade on delete cascade,

constraint fk1 foreign key (ders\_kodu) references ders(ders\_kodu) on update cascade on delete cascade)

burada kolonu parantez içerisine almayı sakın unutma.

constraint fk1 foreign key (ders\_kodu) references ders(ders\_kodu) on update cascade on delete cascade

listemde olan filmlerden satılanları çıkarıyorum satılmayan filmler kalıyor.

küme farkı:

select film\_adi from Film

except

select f.Film\_Adi from Satislar s inner join Film f on f.Film\_No = s.Film\_No

film database’i

create table Film(

Film\_No int identity(1,1) primary key,

Film\_Adi varchar(25) ,

Fiyat int,

Film\_Turu varchar(25),

Film\_Puanı int,

Yonetmen varchar(25),

constraint chk\_puan check(Film\_Puanı>=1 and Film\_Puanı<=10))

create table Musteriler(

Musteri\_ID int identity(202000,1) primary key not null,

Ad varchar(25),

Soyad varchar(25),

Cinsiyet varchar(5) not null,

Dogum\_Tarihi datetime,

constraint chkcins check(Cinsiyet in ('Erkek','Kadın')))

create table Satislar(

Musteri\_no int ,

Film\_No int,

Tarih datetime,

Odeme\_Turu varchar(15),

constraint FK\_FilmNo foreign key (Film\_No) references Film(Film\_No),

constraint FK\_MusteriNo foreign key (Musteri\_No) references Musteriler(Musteri\_ID),

constraint chckOdeme check(Odeme\_Turu in('Kredi Kartı','Nakit')))

select film\_adi from Film

except

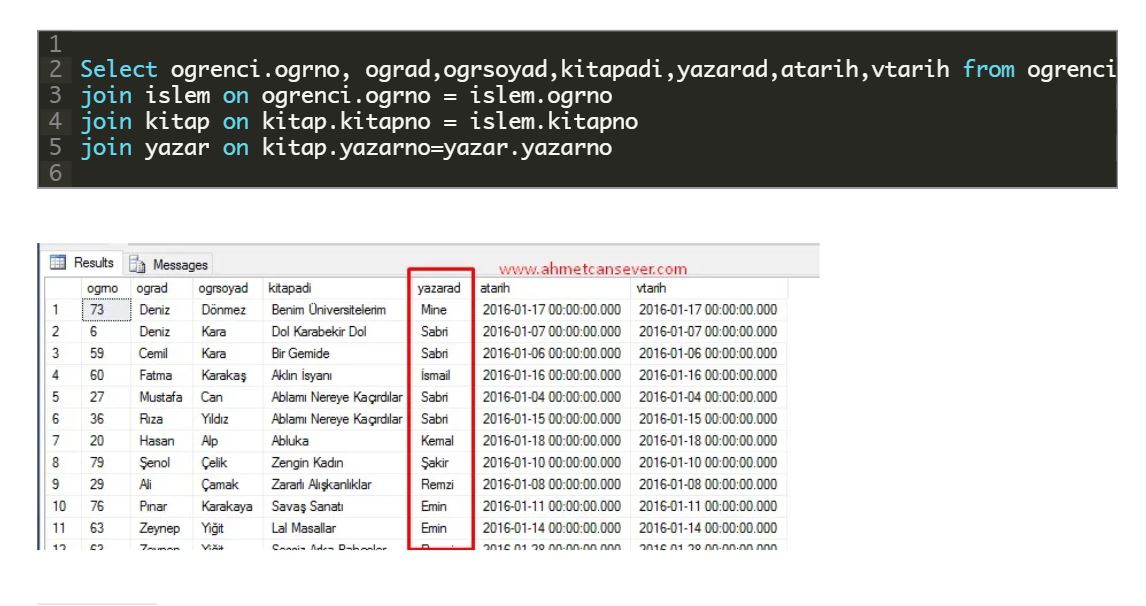
select f.Film\_Adi from Satislar s inner join Film f on f.Film\_No = s.Film\_No

**INTERSECT:** SQL’de INTERSECT ifadesini iki kümenin ortak elemanlarını getirmek için kullanırız.

SELECT Sutun1, Sutun2...., SutunN FROM Tablo1

INTERSECT

SELECT Sutun1, Sutun2...., SutunN FROM Tablo2



Yeni ders

use deneme

alter table ogrenciler add dogum\_tarihi date not null default '1995'

select \* from ogrenciler

alter table ogrenciler add unique(d\_yeri)

alter table ogrenciler add tc\_kimlik int

alter table ogrenciler add constraint unq unique(ogr\_no,tc\_kimlik)

create table ogrenciler1001(

ogr\_no int not null,

adi varchar(25),

soyadi varchar(25),

d\_yeri varchar(20))

alter table ogrenciler1001 add constraint prm primary key(ogr\_no)

alter table ogrenciler1001 add constraint fky foreign key(ogr\_no) references ogrenci(ogr\_no)

alter table ogrenciler1001 add check (ogr\_no>=2019000)

----tablodaki butun degerleri siler

truncate table öğrenciler

order diye fonksiyonum olduğu için bu Order sütununu ya ismini değiştiririm ya da parantez içerisinde yazarım.

select \* from [Order]

select \* from customer

SELECT column\_name

FROM fabrika.INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'customer'

select firstname,lastname,country from Customer

select Customer.City,Customer.FirstName,Customer.Phone from Customer

------ Kartezyen çarpımı olarak doner.

select customer.FirstName as 'MusteriAdi' , Supplier.Country from Customer , Supplier

select \* from Customer

select count(OrderDate) as 'sayı' from [Order]

select count(distinct(OrderDate)) as 'benzersizSayı' from [Order]

select distinct(Package) from Product

select top 5 ıd,CustomerId,TotalAmount from [Order] order by TotalAmount desc

select \* from Customer order by FirstName,country desc

select \* from Customer order by FirstName,country asc

---- sutun numarası vererek de sorgulama yapabiliriz.

select \* from customer order by 5,4 desc

select Id ,city as 'Şehir' ,country as 'Ülke' from Customer

select firstname +SPACE(1) + lastname as 'İsim-Soyisim' from Customer

select Id , ProductName , UnitPrice , UnitPrice\*1.18 as KdvDahil from Product order by UnitPrice desc

ders 21.12.2020

use fabrika

select \* from product where SupplierId=1 AND UnitPrice<22 AND Package='24 - 12 oz bottles'

select \* from Customer where FirstName= 'pa[o]%'

select \* from Customer where FirstName not like 'maria'

select \* from Customer where not FirstName = 'maria'

select \* from Customer where FirstName like '\_[^a]%'

select \* from Customer where FirstName in ('maria','paula','john')

--ismi bunların dışında olanları sıralar

select \* from Customer where FirstName not in ('maria','paula','john')

select \* from OrderItem where UnitPrice>=9 and UnitPrice<=10

select \* from OrderItem where UnitPrice between 9 and 10

select \* from Customer where FirstName like '[a-c]%'

select \* from Customer where FirstName like '%a%'

select \* from Customer where FirstName like 'a%'

select \* from Customer where FirstName like '%a'

select \* from Customer where FirstName like '\_\_\_\_'

select \* from Customer where FirstName like '[l,e]%'

select \* from Customer where FirstName like '[a-d]%'

select \* from Customer where Id like '[2-3][0-9]'

select \* from Customer where Id between 20 and 39

select \* from Customer where FirstName like '[^a,^b]%'

select \* from Customer where FirstName like '[^a-b]%'

select \* from Customer where FirstName not like '[a,b]%'

select \* from Supplier where fax is null

select \* from Supplier where fax is not null

select \* from Customer

set identity customer on ---identity varsa ekleme yapar

set identity customer of -- dışarodan veri girişini kapatır.

insert into customer values('Harun','BAKIRCI','Turkey','İstanbul','0538 635 83 97')

select \* from Customer where FirstName like 'a%'

select \* from Customer where FirstName like '%a'

select \* from Customer where FirstName like '\_\_\_\_'

select \* from Customer where FirstName like '[l,e]%'

select \* from Customer where FirstName like '[a-d]%'

select \* from Customer where Id like '[2-3][0-9]'

select \* from Customer where Id between 20 and 39

select \* from Customer where FirstName like '[^a,^b]%'

select \* from Customer where FirstName like '[^a-b]%'

select \* from Customer where FirstName not like '[a,b]%'

select \* from Supplier where fax is null

select \* from Supplier where fax is not null

select \* from Customer

set identity customer on ---identity varsa ekleme yapar

set identity customer of -- dışarodan veri girişini kapatır.

insert into customer values('Harun','BAKIRCI','Turkey','İstanbul','0538 635 83 97')

select \* from Customer

update customer set City='Barcelona' ,country='Spain' where City='Turkey'

select \* from Supplier

select \* from Customer

update Supplier set Fax='Belirtilmemiş' where Fax is null

select ISNULL(fax,'Belirtilmemiş') as Fax from Supplier

delete from Customer where RIGHT(firstname,2) =

--- isminin son iki harfi un olanları sil

delete from Customer where FirstName like '%un'

insert into customer values('Harun','BAKIRCI','Turkey','İstanbul','0538 635 83 97')

select \* from Customer

--- sildikten sonra ıd numarasını baştan başlatmaz en son sildiğin id ne ise onunla devam eder

---or 95 id li en kayıdı sildiniz yeni kayıt 96 olarak algılanır bunu önlemek için truncate ekle

---delete from customer id numarası gibi kalıntıları hatırlar

--- truncate customer dersek tablodaki herşeyi siler

use fabrika

select Country ,city , count(ıd) as MusteriSayisi from Customer group by Country ,City

select Country ,city , count(City) as MusteriSayisi from Customer group by Country ,City

select Country , count(ıd) as MusteriSayisi from Customer group by Country

select OrderId, count(OrderId) as A from OrderItem group by OrderId having count(OrderId) = 3

SELECT Country, COUNT(Id) as ToplamMusteriSayisi FROM Customer WHERE Country <> 'USA'

GROUP BY Country HAVING COUNT(Id) >= 9 ORDER BY COUNT(Id) DESC

SELECT Country, COUNT(Id) as ToplamMusteriSayisi FROM Customer

GROUP BY Country HAVING COUNT(Id) >= 9 and Country NOT like 'USA' ORDER BY COUNT(Id) DESC

select \* from Supplier

select \* from Customer

select c.Id , s.CompanyName from Customer c inner join Supplier s on c.Id = s.Id

select c.Id , s.CompanyName from Customer c inner join Supplier s on c.Id = s.Id where c.Id>10

select \* from Customer where Customer.Id =(select CustomerId

from [Order] where TotalAmount=6750)

select \* from Customer where Customer.Id =(select CustomerId

from [Order] where TotalAmount=45)

select \* from [Order]

select \* from Customer

select top 1 UnitPrice from Product where Package = '24 - 12 oz bottles' order by UnitPrice

select Package ,UnitPrice from Product where UnitPrice <

(select top 1 UnitPrice from Product where Package = '24 - 12 oz bottles' order by UnitPrice) order by UnitPrice

select \* from Product where UnitPrice <

(select top 1 UnitPrice from Product where Package = '24 - 12 oz bottles' order by UnitPrice)

select \* from [Order] where TotalAmount < (select avg(TotalAmount) from [Order]) order by TotalAmount

---içiçe sorguda kesinlikle () koy

select \* from Product

select \* from Customer

--- birden fazla müşteri oldugu için in kullandım.

select \* from Customer where Customer.Id in

(select [Order].CustomerId from [Order] where TotalAmount > (select avg(TotalAmount) from [Order]))

select \* from Customer where Customer.Id in

(select Product.Id from Product where Product.ProductName='Ikura')