Tablo oluştururuken Kullanılan Komutlar :

· Create database name

· Create table name

· Use name

· Not null

· İnt, varchar,time,smalltime,money,bit(0 yada 1)

· (isim) int primary key

· İdentity (1,1) (oduncno int primary key not null IDENTITY(1,1),)

· default ('adet')//varsayılan değer

· getdate //datetime default getdate()

Tabloyu Oluşturduktan Sonra Kullanılcak Komutlar :

· alter table musteri alter column musteriad Varchar(20) //müşteri adlı kolonu varchar (50) i yapar

· alter table musteri alter column musteriid int not null //musteri tablosundaki musteriid yi not null a çevirir

· alter table musteri add email varchar(30) //email adlı kolon ekler

· alter table musteri add constraint in\_cins check (cinsiyet in('E','K')) //cinsiyete sadece e k harfleri girilsin

· alter table musteri add constraint chk\_tc check (len(tcno)=11) //tcno hücerlerine 11 karakter girilsin

· alter table musteri add constraint pk\_musteriid primary key (musteriid) //müşteriid kısımını birincil anahtar yapar

· insert into musteri values ('2','kadir','abatay','11111111112','a','5435','E','7.9.2000')

· alter table siparis\_detay add toplamfiyat as(miktar\*satisfiati) //toplam fiyat siye kısım ekler ve eklerken miktarla satis fiyarını çarpar

· alter table yazar drop column yas //yazar tablosundaki yaş kısmını kaldırır

· alter table kitap add constraint eşşiz\_ISBNNo unique (ISBNNo) // ISBNNo kısmına aynı veri girişi yapılamaz

· EXEC SP\_RENAME ‘kitap’,’kitaplar’; //kitap tablosunun adını kitaplar yapar

İlişki Kurma :

· alter table siparis add constraint fk\_asd foreign key ( musteriid) references musteri(musteriid) //siparis tablosundaki musteriid kısmını musteri tablosundaki musteriid kısmı ile ilişkilendirir musteriàsiparis ilişkinin adı fk\_asd olur

· alter table siparis drop constraint fk\_asd //fk\_asd ilişkisini kaldırır

Kayıt Güncelleme :

update personel set soyad='DEMİR' where ad='EMEL' and soyad='YILMAZ'

update personel set maas=maas+maas\*0.05 where kıdem>10

update personel set kıdem+=1

update personel set bolum='SATIŞ' where ad='Ahmet'

update personel set maas=maas+250 where maas<2500

Kayıt Silme :

delete from personel where ad='Emine'

Tabloda Bilgi Arama :

· select \* from uye //uye tablosunu getirir

· select kitapadı,kitap\_özeti from Kitap //kitap tablosundaki

kitapadı,kitap\_özeti ni getirir

· select \* from Kitap where sayfasayisi>100 //sayfa sayısı 100 den büyük olanları getirir

·

· select \* from yazar where adi like'k%' //yazar adı k ile başlayanlar gelir

·

· select \* from personel where d\_tarih>’01.01.1977 ‘//doğum tarihi belirtilen değerlerden büyük olan değerleri getirir

·

· select \* from personel where soy\_ad like '\_\_Ş%'//üçüncü harfi ş olan

·

· select \* from personel where maas>3500 and cinsiyet='BAYAN' and bolum != 'SATIŞ' //maaşı 3500 den büyük olup bayan ve satış departmanından çalışmayanlar içilen geçerli

·

· select \* from personel where cinsiyet='BAY' and bolum ='SATIŞ' or bolum='Muhasabe' //cinsiyeti

·

· select \*from personel where ad like '%A%' and kıdem between 5 and 9 or bolum IN('Satış','Muhasebe') or cinsiyet='BAYAN' //adının iç,inde a olup kıdemi 5 ile 9 arasında olup bayan ve bölümü ya satış ya da muhasebe olan

·

· select Top(5)\*from personel order by maas desc //azalan maaş a göre sırlama yapıp ilk 5 şi gösterir

· select Top(3)\*from personel order by kıdem asc //artan kıdem yapıp ilk üçü gösterir

·

· select \* from Musteriler where MusteriID like '[A-F]% //MusteriID harfleri A ile F ile arasında başlayanları getirir

·

· select \* from Personeller order by [Unvan],[Adi]

· select Top(5) \* from urunler order by [BirimFiyati] DESC

· select Sehir from Personeller group by Sehir //şehirleri grup layıp her bir şehiri bir seferde gösterir

--Count,MAX,MIN,AVG,SUM :

select Count (\*) as 'pesonel sayısı' from personel //personel tablosundaki kayıt sayısı

select Avg (maas) as 'Ortalama maas' from personel

select Min (maas) as 'Endüşük Maaş' from personel

select Sum (maas) as 'Toplam Maaş' from personel

select bolum,COUNT(\*) as 'Çalışan Sayısı' from personel group by bolum // her bölümdeki çalışan sayısı

select bolum,Sum(maas) as ' Toplam Maas' from personel group by bolum

//her bölümdeki toplam maaş ları hesaplar

select cinsiyet,COUNT(\*) as 'Çalışan Sayısı' from personel group by cinsiyet

//cinsiyet lerdeki insan çalışan sayısı

select cinsiyet,Min(kıdem) as 'Kıdem Sayısı' from personel group by cinsiyet

//cinsiyetlerdeki en düşük kıdemleri gösterir

select bolum,cinsiyet,COUNT(\*) as 'Çalışan Sayısı' from personel group by cinsiyet, bolum

//cinsiyetle ve bolumü gruplayıp çalışan sayılarına bulur

SELECT SUM(Total) from Invoice WHERE InvoiceDate LIKE '%2008%'; //2008 yılına ait satılan albümlerin fiyat toplamını hesaplıyor

SELECT SUM(Total) from Invoice WHERE InvoiceDate LIKE '%2009%’;

SELECT SUM(Total) from Invoice WHERE InvoiceDate LIKE '%2010%';

—————————————————————

SELECT A.Title, T.Name //albüm ve track tablolarını birleştirerek tablolardaki

FROM Album A JOIN Track T AlbumId leri aynı olan satırları verir

On T.AlbumId=A.AlbumId

—————————————————————

SELECT A.Title, T.Name ,Ar.Name ,T.UnitPrice

FROM Album A JOIN Track T JOIN Artist Ar

On T.AlbumId=A.AlbumId and Ar.ArtistId =A.ArtistId

—————————————————————

SELECT A.Title, T.Name ,Ar.Name ,mt.Name

FROM Album A JOIN Track T JOIN Artist Ar JOIN MediaType mt

On T.AlbumId=A.AlbumId and Ar.ArtistId =A.ArtistId and mt.MediaTypeId =T.MediaTypeId

—————————————————————

DECLARE @sayı INT //sql while kullanımı

SET @sayı=1

WHILE ( @sayı <= 10)

BEGIN

PRINT @sayı

SET @sayı = @sayı + 1

END

———————————————————

SELECT DISTINCT AlbumId from Track t //Track tablosundaki Albumıd lerini tekrarlamadan getirir.

———————————————————

SELECT Bytes from Track t Order BY Bytes //track tablosundaki bytes alanını büyük ten küçüğe sıralama yapar.

———————————————————-

SELECT \* FROM SalesLT.Product WHERE Color ='Red' // SalesLT.Product tablosundaki color sutunda red olanları seçer.

———————————————————-

SELECT \* FROM SalesLT.Product WHERE Color ='Red' and [Size] ='44’// SalesLT.Product tablosundaki color sutunda red ve size 44 olanları seçer.

———————————————————-

SELECT \* FROM SalesLT.Product WHERE Color ='Red' or Color='Blue’//or(veya) SalesLT.Product tablosundaki color'ı Blue ve red olanları seçer.

————————————————————

SELECT TOP 1 \* FROM SalesLT.Address;//tablodaki ilk kayıdı seçer.

———————————————————

SELECT \* FROM SalesLT.Address where city LIKE '[B,E]%' ;//tablodaki Şehir isimleri B ve E le başlayanları getirir

———————————————————

SELECT \* FROM SalesLT.Address where city LIKE '[B-E]%' ;//tablodaki Şehir isimleri B ve E arasındaki harflerle başlayanları(B ve E dahil) getirir.

———————————————————

SELECT \* FROM Personel WHERE Sehir LIKE 'İzmi\_' //Bu kod Sehir alanının İzmi ile başlayan ve son harfi ne olursa olsun farketmeyen kayıtları seçer.

———————————————————

IN bir operatördür ve WHERE ile kullanılır.

SELECT \* FROM Personel WHERE Sehir IN ( 'İstanbul','Kocaeli')//Sehir alanında İstanbul ve Kocaeli yazan kayıtlar seçilir.

———————————————————

BETWEEN bir operatördür ve WHERE ile kullanılır.

select \* from SalesLT.CustomerAddress where AddressID Between 500 and 600 //SalesLT.CustomerAddress adres adres tablosundaki AddressID'si 500 ile 600 arasında olanları seçer.

SELECT \* FROM SalesLT.Address where PostalCode between 'k' and 'z' ;//SalesLT.Address tablosundaki postalcode k ve z arasındaki kelimelerle başlayanları getirir.

SELECT \* FROM SalesLT.Address where PostalCode not between 'k' and 'z' ;

/SalesLT.Address tablosundaki postalcode k ve z arasındaki kelimelerle başlamayanları getirir.

———————————————————

select city as sehir from SalesLT.Address// bu tablodaki city alanına as kullanaraka geçici olarak sehir ismi verirlir.

.———————————————————

UNION and. UNION ALL

SELECT ArtistId from Artist // artist ve album tablolarındaki artistıd lerini

UNION tekrarlamadan ve sırayla getirir.

select ArtistId from Album

SELECT ArtistId from Artist //iki tablodaki artistıd lerini olduğu gibi birleştirir

UNION ALL

select ArtistId from Album

———————————————————

NOT:Select Into

Insert Into Select

———————————————————

CREATE TABLE Personel //personel isimli tablo oluşturur

(

id INT,

adi\_soyadi varchar(25),

sehir varchar(15),

bolum varchar(15),l

)

———————————————————

ALTER TABLE Cay\_Ocakları DROP COLUMN id;//cay ocağı tablosunda id sütununu kaldırır

———————————————————

delete from Urunler where Fiyatı = 10

———————————————————

DROP DATABASE veritabani\_adi; // veri tabanını siler

———————————————————

update Nitelik //Nitelik\_id si 10 olan verinin nitelik kısmını

set Nitelik='Normal' normal olarak değiştirir

where Nitelik\_id=10

———————————————————

/\*

>delete +

>update +

>Count + ,MAX + ,MIN +,AVG,SUM

>Date +,Getadate +

>inner join,left join,right join

>having,group by +

>iç içe select

\*/

select \* from Urunler

select \* from Personeller

select \* from [Satis Detaylari]

----delete

delete from Urunler where UrunID=1

delete from Urunler where UrunAdi='Chai'

----update

update Urunler set UrunAdi='a' where UrunAdi='Chang'

update Urunler set KategoriID=3 where UrunAdi=8

-----Count + ,MAX + ,MIN +,AVG,SUM

select Sum(u.BirimFiyati) as 'ürünlerin Birim Fiyatlarının Toplamı' from Urunler u

select Min(u.BirimFiyati) as 'en pahalı ürün' from Urunler u

select u.UrunAdi,Max(u.BirimFiyati) as 'en pahalı ürün' from Urunler u group by UrunAdi ( select top(1) u.UrunAdi,Max(u.BirimFiyati) as 'en pahalı ürün' from Urunler u group by UrunAdi order by Max(u.BirimFiyati) desc )---bak

select COUNT(u.UrunID) from Urunler u

select Count (\*) as 'pesonel sayısı' from personeller p

select COUNT(\*) as 'kategoridaki adet sayısı'from Urunler u group by u.KategoriID

-----Date,Getadate

select Top(1) IseBaslamaTarihi ,year(GETDATE())-YEAR(IseBaslamaTarihi) as 'Çalıştığı Yıl sayısı' from Personeller order by IseBaslamaTarihi asc

select p.Adi, year(GETDATE())-YEAR(DogumTarihi) from Personeller p

----group by , having

select u.KategoriID,avg(u.BirimFiyati)

from Urunler u group by u.KategoriID having avg(u.BirimFiyati) >25 ---Her bir kategorideki fiyatların ortalamasını bul ve hangi kategorinin

--birim fiyat ortalaması 25 büyük olsun

select u.KategoriID,sum(u.BirimFiyati) from Urunler u group by u.KategoriID having sum(u.BirimFiyati)>200

select Sehir,avg(YEAR(GETDATE())-YEAR(DogumTarihi)) from Personeller group by Sehir having avg(YEAR(GETDATE())-YEAR(DogumTarihi))>60

----Joinler

select t.MusteriAdi,u.UrunAdi from urunler u inner join Tedarikciler t on u.TedarikciID=t.TedarikciID

select p.Adi,UrunAdi from Personeller p inner join Satislar s on p.PersonelID=s.PersonelID

inner join [Satis Detaylari] sd on sd.SatisID=s.SatisID

inner join Urunler u on sd.UrunID=u.UrunID group by u.UrunAdi , p.Adi

UPDATE Urunler set UrunAdi ='x' where KategoriID=(select KategoriID from Kategoriler where KategoriAdi='Beverages')

delete from Urunler where TedarikciID=(select TedarikciID from Tedarikciler where MusteriAdi='Beate Vileid' )