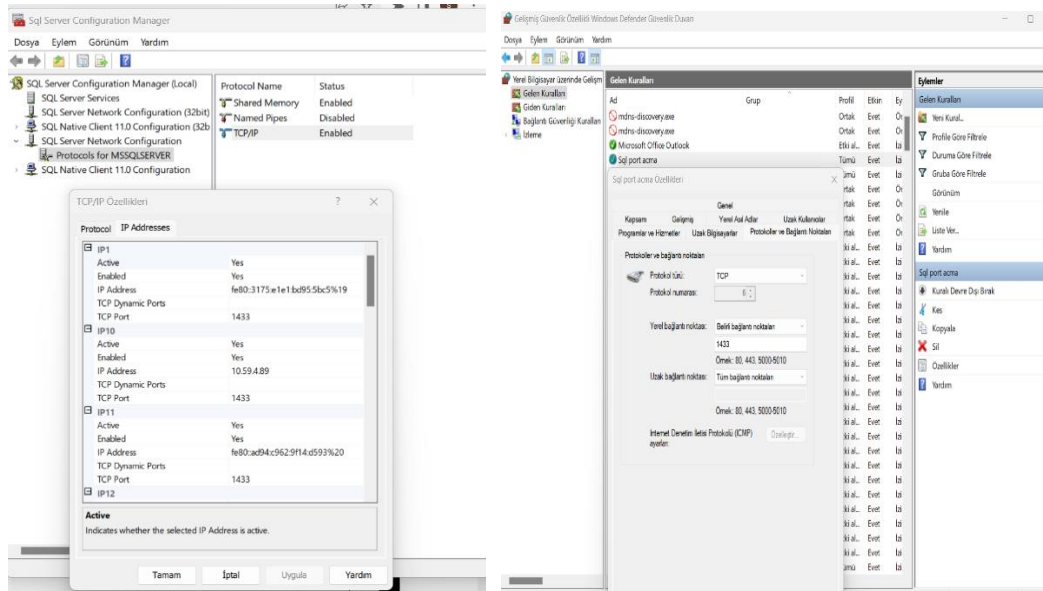


KISIM:

TARİH: 07/07/2023

## YAPILAN İŞ: Sql

- Bana kullandıkları MES sistemini ,nasıl çalıştığını anlattılar. MES sisteminin üretimde kullanılan her türlü metot ile aracın toplandığı, çevrimiçi olarak entegre edilen bilgisayar destekli bir sistem olduğunu öğrendim, MES sistemiyle nasıl çalıştıklarını öğrendim.
- Öncelikle sql ile çalışmaya başladım. Sql server management studio'nun kurulumunu daha önce tamamlamıştım. Bu nedenle arayüze hakim olmak amacıyla yeni sorgu yazmayı, tablo oluşturup içerisine veri eklemeyi, sorguyu çalıştırma gibi temel işlevleri öğrendim.
- Sql'de port açmayı öğrendim. Sql portunun uzak bağlantı kurmak için kullanıldığını öğrendim. Öncelikle SQL Server Configuration Managerdaki TCP/IP 'yi enabled yaparak bağlantı kabul edilmesini aktifleştirdim. IP adreslerini hepsini 1433 yaptım ve izin verdim. Sunucuyu tekrar başlattım. Sonrasında Windows Firewall ayarlarını düzenledim ve 1433/TCP potunu aktifleştirdim. Yeni kural oluşturdum. Ayarlarını düzenlerken kendi IP adresimi ve 1433 portunu ekledim. <https://portchecker.co/checking> bu siteden bağlantı noktasının açılıp açılmadığını kontrol ettim. Bu adımları tamamladıktan sonra sql server management studio'da connect to server penceresinde sunucunun IP adresini, kullanıcı adını ve şifresini girerek bağlanmış oldum.



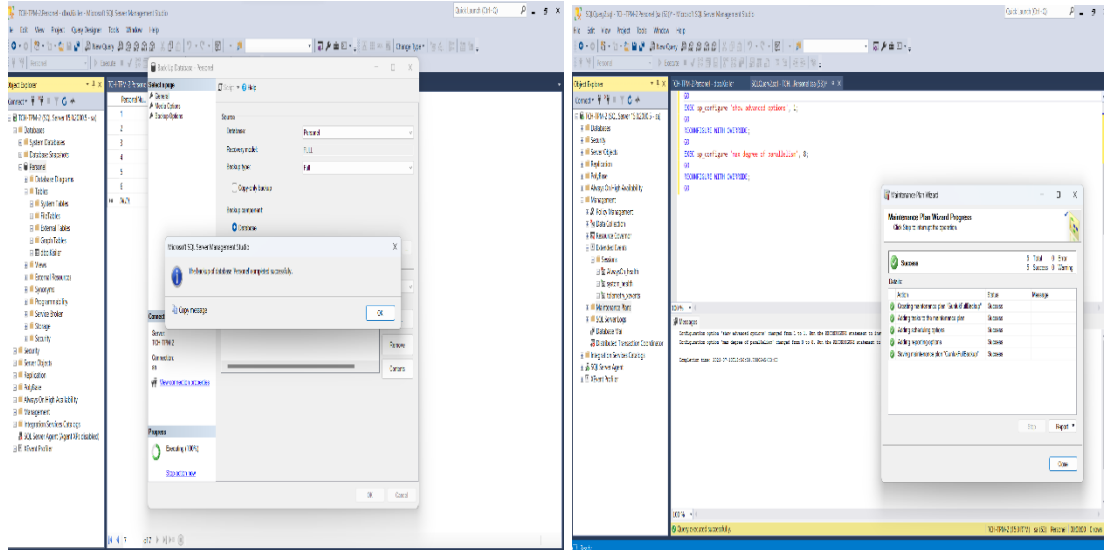
KONTROL SONUCU:

KISIM:

TARİH:10.07.2023

## YAPILAN İŞ: Sql

- Sql de backup almanın veritabanının bozulma ihtimaline karşı yedek alma işlemi olduğunu öğrendim. Backup almayı, backup yüklemeyi ve otomatik backup ın nasıl alındığını öğrendim.
- Sql de backup almak için öncelikle management studioyu açtıktan sonra databases genişletilir ve yedeğini alacağımı veritabanının üzerine sağ tıkladım. Açılan menüde Tasks + Sql Server backup'a tıkladım. Sonrasındaki pencerede Sql yedek alma tipini tam yedek alma olarak belirttim. Add butonuna tıklayarak yedeğin alınacağı dosya yolu seçtim. Alacağım yedeğe isim de verdikten sonra ilerledim ve yedek alma işlemini tamamladım. Sql de backup yüklemek için SSMS'te herhangi bir veritabanı üzerinde sağ tık yaparak açılan menüde tasks+restore+database ' tıkladım. Daha sonrasında açılan pencerede devicea tıkladım ve yedeğin bulunduğu yeri seçtim. Böylece yedeği geri yüklemiş oldum.
- Sql de otomatik backup almanın manuel alınan backup'ların belirttiğimiz gün ve saatlerde otomatik olarak alınmasını sağladığını öğrendim. Otomatik backup almak için SSMS' te management+management pşans+ maintenance plan wizard'a tıkladım. Daha sonrasında backupa isim verdim. Tek bir görev oluşturacağım için single schedule seçeneğini seçtim. Change butonuna tıklayarak zaman alanını düzenledim. Sonrasında backup database(Full) seçtim. Otomatik backup alınacak database'leri seçip Disk'i işaretledim. Kaç gün sonra silineceğini belirledim. Log dosyalarının tutulacağı yeri seçip tamamladım ve başarıyla oluştuğunu gördüm. Oluşturduğum bu otomatik backup'ı maintenance plans 'ın altında görüp execute diyerek çalıştırdım ve işlemin başarıyla çalışıp kaydedildiğini gördüm.



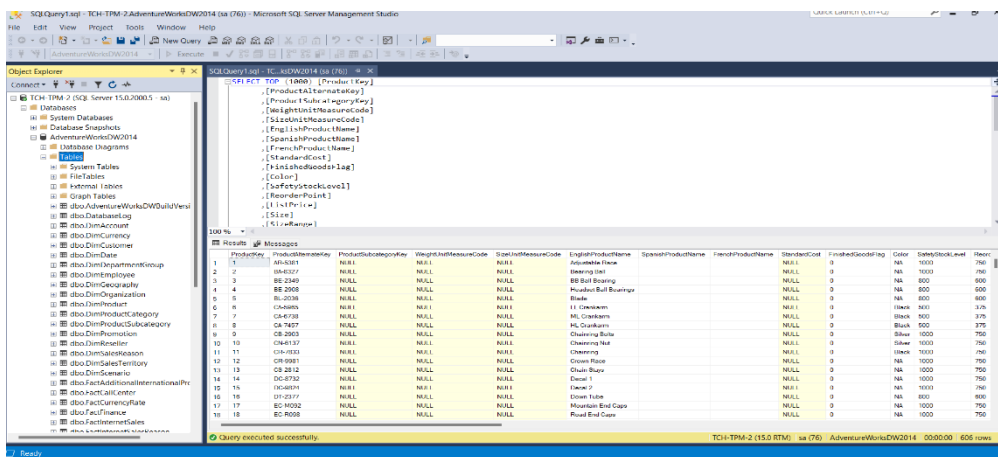
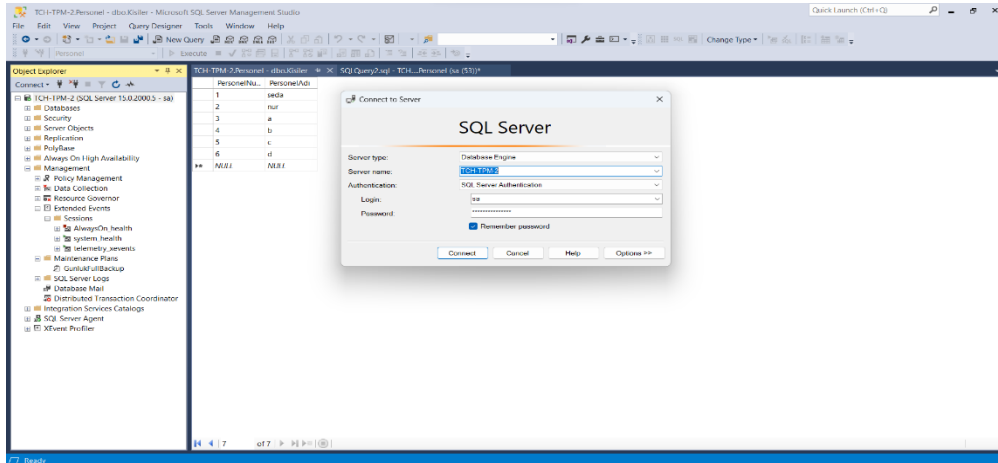
KONTROL SONUCU:

KISIM:

TARİH:11.07.2023

## YAPILAN İŞ: Sql

- Sql'de 'sa' (system administrator) kullanıcısının ve şifresinin önemini, nasıl aktif edileceğini öğrendim. 'sa şifresinin' veritabanının güvenliğini sağladığını öğrendim. Sql'de 'sa' kullanıcısının ve şifresinin önemini, nasıl aktif edileceğini öğrendim. 'sa' kullanıcısında properties->status'a gittim. Grant ve enabled 'i' seçtim. General'de şifreyi yeniledim. Sonrasında sql'e sağ tık ile properties'e gittim. Sql Server and Windows Authentication mode'u seçtim ve Sql Server configuration manager 'dan restart yaptım. Yeniden başlatıp sa kullanıcısı ile giriş yaptım.
- AdventureWorks Database ücretsiz bir database örneğidir. Kurulumu yapabilmek için öncelikle database dosyasını indirdim. İndirdiğim .bak uzantılı dosyayı C:\ dosyasına taşıdım. Daha sonra Microsoft SQL Server Management Studio'da database->restore database'e tıkladım. Device'da .bak uzantılı indirdiğim dosyayı ekledim ve tablolar geldi.



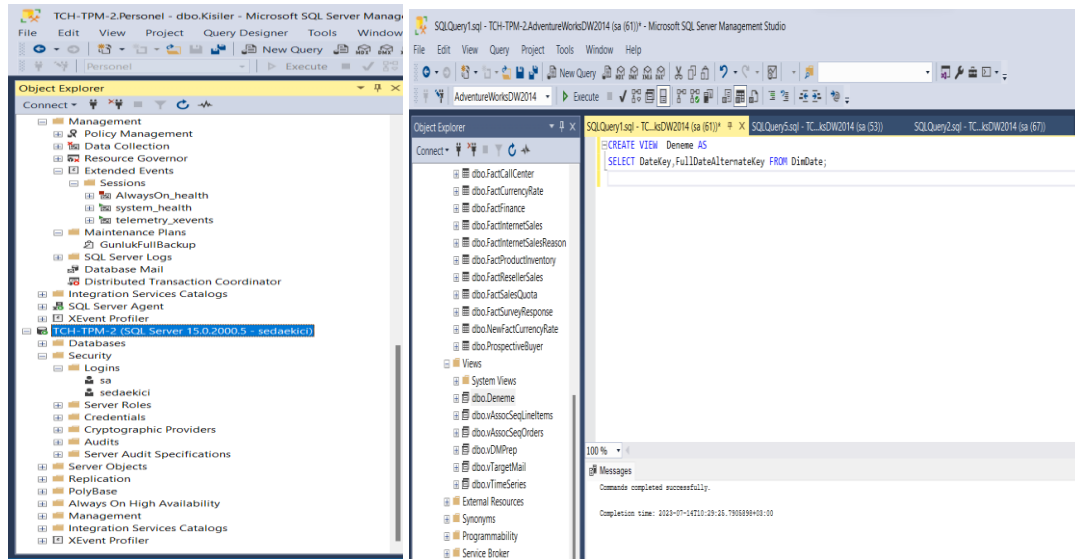
KONTROL SONUCU:

KISIM:

TARİH:12.07.2023

## YAPILAN İŞ: Sql

- User, Database bazlı kullanıcı oluşturma şeklidir. Kullanıcı User'ı oluşturulduğunda Database üzerinde işlemler yapabilmesidir. Login kullanıcıları, tüm veri tabanlarına erişebilir ve belirlenen yetki kapsamlarında çalışabilirken, User kullanıcılar sadece belirli veri tabanlarında, belirlenen yetki kapsamlarında çalışabilirler. Sql serverda login kullanıcı oluşturmak ve yetki vermek için security-> login-> new login seçeneğini seçtim. İsim verip sql server authentication seçeneğini seçip şifre belirledim. Daha sonra rolleri belirledim. Burada seçebileceğim roller bulkadmin (Bulk Insert Administrator – Çoklu Kayıt Yöneticisi), dbcreator (Database Creator – Veri tabanı Yöneticisi), diskadmin (Disk Administrator – Dosya Yöneticisi), processadmin(Process Administrator – İşlemci Yöneticisi), public (Herkese Kısıtlı Hak), securityadmin (Security Administrator – Güvenlik Yöneticisi), serveradmin (Server Administrator – Server Yöneticisi), setupadmin (Setup Administrator), sysadmin (System Administrator – Sistem Yöneticisi)'dir. User Mapping alanında oluşturacağımız Login in hangi veri tabanlarına erişiminin ve bu veri tabanlarında hangi yetkilerinin olacağını belirledim. Connect SQL e abğlanma yetkisi verdim. Grant ve enabled 'i seçtim ve kullanıcıyı oluşturdum. Sql server'a giriş ekranında kullanıcı adı ve şifreyi girerek giriş yaptım.
- Sql view oluşturma database için hazırladığım sorgunun daha sonra kullanılmak amacıyla saklanması sağladığını öğrendim. Gerçek bir tablo değildir ancak tablo gibi kullanılabilir. View oluşturmak için CREATE VIEW komutundan sonra bir isim vererek AS komutu sonrasında sorgumu yazarak kullandım.



KONTROL SONUCU:

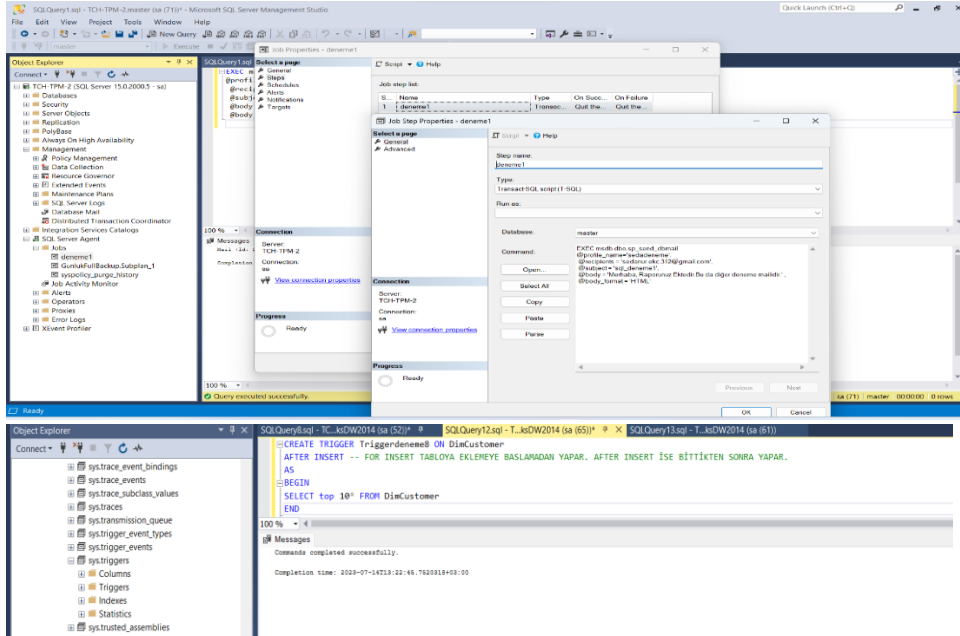


KISIM:

TARİH:14.07.2023

## YAPILAN İŞ: Sql

- SQL Server’da Job oluşturmayı öğrendim. SQL Server’da Job’lar SQL Server Agent servisi aracılığıyla pek çok işlemi sırasıyla belirlenen periyotlarla gerçekleştirmeyi sağlar. Bir kez veya daha sonra tekrar kullanabiliriz. Oluşturduğumuz jobların başarıları durumunu da izleyebiliriz. Job oluşturmak için Sql Server Agent ->New-> Job’a tıkladım. Gelen sayfada Job’a isim verdim. Owner kısmına da sa kullanıcıasını seçtim. Steps kısmında new diyerek isim verdim ve T-Sql ‘i seçtim. Command kısmına ise gmail atmayla ilgili yazdığım kodları ekleyip kaydettim ve ilgili olduğu database’i seçtim. Advenced sayfasında oluşturduğum bu adım bittiğinde başarılı çalışması sonucunda raporlanmasını istedim. Schedules kısmında Job’un hangi aralıklarla çalışmasını istediğimi düzenledim. Alarm oluşturmadım. Targets kısmında target local server ‘ı işaretledim ve tamamladım.
- SQL Server’da Triggers oluşturmayı öğrendim. Bir tabloda ekleme, güncelleme ve silme işlemlerinden biri gerçekleştiğinde veya gerçekleşmeden önce, aynı tabloda veya başka bir tabloda belirli işlemlerin otomatik yapılmasını sağlar. Create trigger komutundan sonra triggera isim verdim ve on ile hangi tabloda çalışacağını belirttim. After insert komutuyla tabloya ekleme yaptıktan sonra işlemleri yapmasını sağladım. As komutundan sonra begin ve end komutları arasında yapmak istediğim işlemleri yazdım ve triggers oluşturdum.



KONTROL SONUCU:



KISIM:

TARİH:17.07.2023

YAPILAN İŞ: Sql

- Sql Management Studio’da sorgu yazma üzerine çalıştım. Komutları ve nasıl kullanıldığını öğrendim.
- SELECT, LEFT JOIN, SUBQUERY, GROUP BY, HAVING,ORDER BY,SUM,MAX,MIN,COUNT,ÜST TABLO,WITH(NO LOCK),ROLLBACK,TOP,UNION ALL,UNION, FUNCTION,MONTH, YEAR,GETDATE,DATEDIFF,DATEADD,CASE WHEN,CONVERT,CAST,ROUND,FLOOR,ISNULL,DISTINCT, WHERE,OPERATORLER(<,>=,BETWEEN),LIKE,NOT LIKE,IN,OR,INSERT INTO,CURSOR,COLLATE,UPDATE,DELETE,EXISTS,ALTER,INSERT INTO SELECT, REPLACE,DATEPART vb. komutların kullanımını öğrendim.
- AdventureWorksDW2014 deki tablolarla ilgili sorgulara çalıştım.

The first screenshot shows a SQL query in the 'Query Editor' window of SQL Server Enterprise Manager. The query is a SELECT statement with a subquery. The query is:   
SELECT \* FROM DimProductCategory  
--SELECT EnglishProductCategoryName FROM DimProductCategory  
SELECT SalesAmount FROM FactInternetSales ORDER BY SalesAmount DESC

The second screenshot shows the results of a query in the 'Results' window. The query is:   
SELECT \* FROM DimProductCategory  
SELECT EnglishProductCategoryName FROM DimProductCategory  
SELECT SalesAmount FROM FactInternetSales ORDER BY SalesAmount DESC

The third screenshot shows the results of a query in the 'Results' window. The query is:   
SELECT \* FROM DimProductCategory  
SELECT EnglishProductCategoryName FROM DimProductCategory  
SELECT SalesAmount FROM FactInternetSales ORDER BY SalesAmount DESC

KONTROL SONUCU:

Kocaeli Üniversitesi Mühendislik Fakültesi

KISIM:

TARİH:18.07.2023

## YAPILAN İŞ: Sql

- Sql Management Studio’da sorgu yazma üzerine çalıştım. Komutları ve nasıl kullanıldığını öğrendim. select, left join, subquery, group by, having, order by, sum, max, min, count, üst tablo, with(no lock), rollback, top, union all, union, function, month, year, getdate, datediff, dateadd, case when, convert, cast, round, floor, isnull, distinct, where, operatorler(<>, =, between), like, not like, in, or, insert into, cursor, collate, update, delete, exists, alter, insert into select, replace, datepart vb. komutların kullanımını öğrendim.
- Şirket adında yeni bir database üzerinden sorgulara çalışmaya devam ettim.
- Sql’de normalizasyona çalıştım. Veritabanlarında çok fazla sütun ve satırdan oluşan bir tabloyu tekrarlardan arındırmak için daha az satır ve sütun içeren alt kümelerine ayırma işlemi olduğunu öğrendim. Nasıl kullanıldığını ve önemini öğrendim.

The screenshot displays the SQL Server Enterprise Manager interface on the left, showing the 'Sirket' database structure with tables like 'Personel', 'BakimlaYukumu', and 'Proje'. The main window shows a SQL query in the 'SQL Query' editor, which is a complex query involving multiple joins and aggregations. The query is as follows:

```
SELECT P.BolumID, B.BolumAd, COUNT(*) PER_SAYISI
FROM Personel P
LEFT JOIN Bolum B ON P.BolumID=B.BolumID
GROUP BY P.BolumID, B.BolumAd
HAVING COUNT(*)>=3

select p.BolumID, b.BolumAd, count(*) PerSay
from Personel p
inner join Bolum b on p.BolumID=b.BolumID
where p.BolumID in (select p2.BolumID
from Personel p2
group by p2.BolumID
having count(*)>=3)
and p.Maas>500
group by p.BolumID, b.BolumAd
select p.PerID, p.Ad + ' ' + p.Soyad [Ad Soyad],
b.BolumAd, sum(pp.CalismaSaat) TopSaat,
count(*) ProjeSay, 1.0*sum(pp.CalismaSaat)/count(*)
OrtSaat
from Personel p
inner join Bolum b on p.BolumID=b.BolumID
inner join PersonelProje pp on p.PerID=pp.PerID
where p.PerID in (select bk.PerID
from BakimlaYukumu bk
group by bk.PerID
having count(*)>=2)
group by p.PerID, p.Ad + ' ' + p.Soyad, b.BolumAd
order by b.BolumAd, OrtSaat desc
```

The bottom right window shows the results of the query, displaying a table with columns: Ad, Soyad, PerID, DogumTarihi, Adres, Cinsiyet, Maas, AnvID, BolumID. The results are as follows:

Ad	Soyad	PerID	DogumTarihi	Adres	Cinsiyet	Maas	AnvID	BolumID
Ali	Yılmaz	101	1974-01-01 00:00:00	Pasta Sok 11 Kadıköy	E	900	NULL	1
Ali	Yılmaz	101	1974-01-01 00:00:00	Pasta Sok 11 Kadıköy	E	900	NULL	1
Ali	Yılmaz	101	1974-01-01 00:00:00	Pasta Sok 11 Kadıköy	E	900	NULL	1
Zeynep	Demir	202	1972-03-01 00:00:00	Dosano	K	1000	101	3
Zeynep	Demir	202	1972-03-01 00:00:00	Dosano	K	1000	101	3
Ahmet	Kartal	505	1981-04-04 00:00:00	Beşiktaş	E	800	101	5

KONTROL SONUCU:

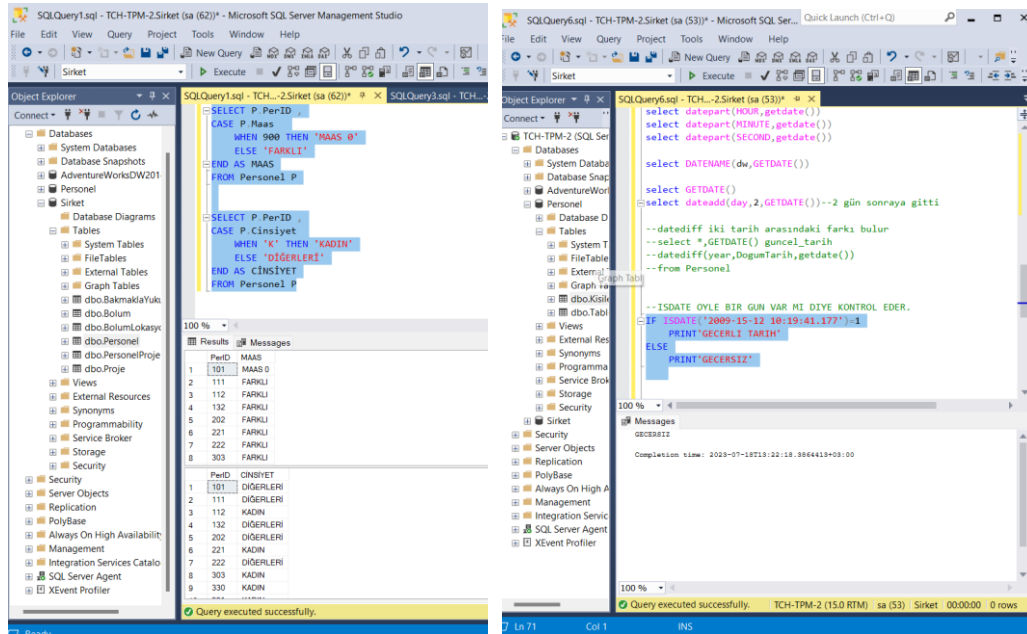


KISIM:

TARİH:19.07.2023

## YAPILAN İŞ: Sql

- T-Sql'de Case When kullanımını, cast işlemini, date fonksiyonlarını ve nasıl kullanıldığını, T-Sql'de fonksiyon oluşturmayı, round ile sayı yuvarlamayı öğrendim. Elimizdeki veriyi koşullara göre yeniden düzenleyebilmek ve sorgulayabilmek için SQL'de CASE WHEN ifadesinin kullanıldığını, cast fonksiyonunun veri tipi dönüşümünde kullanıldığını, getdate() gibi fonksiyonlarının tarih işlemlerinde kullanıldığını, creat komutu ile fonksiyon oluşturulduğunu, round komutu ile sayıları yuvarlamayı öğrendim.
- Data type'a çalıştım, ne olduğunu öğrendim. SQL'de veri tipinin, bir sütunun veya değişkenin depolayabileceği veri türünü belirttiğini; Veri türlerinin, verilerin doğru şekilde depolanmasını ve işlenmesini sağlamaya yardımcı olduğunu öğrendim. SQL'de kullanılan bazı yaygın veri türleri şunlardır: INT'ın tam sayılar için kullanıldığını, VARCHAR'ın metinler için kullanıldığını, DATE'in tarihler için kullanıldığını, TIME'ın saatler için kullanıldığını, TIMESTAMP'ın tarih ve saatler için kullanıldığını, DECIMAL'in ondalık sayılar için kullanıldığını, BLOB'un büyük veriler için kullanıldığını öğrendim.
- Veri türlerini seçerken, depolanacak verilerin türüne dikkat etmek önemli olduğunu, veri türlerini doğru seçmenin, performansı etkilediğini öğrendim.



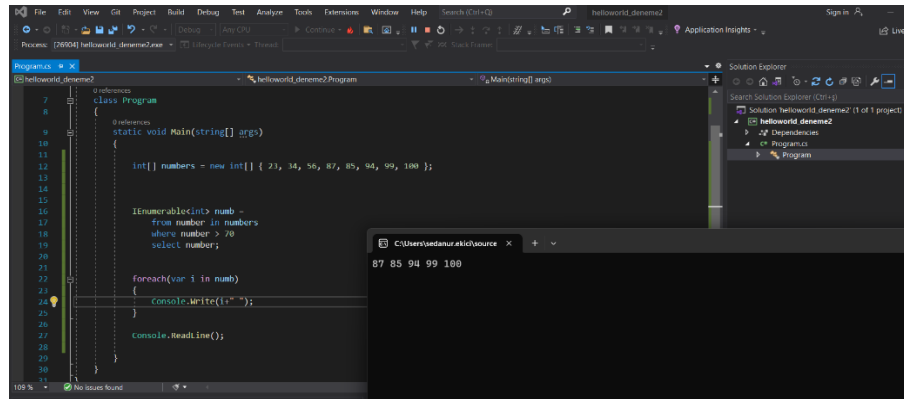
KONTROL SONUCU:

KISIM:

TARİH:20.07.2023

## YAPILAN İŞ: Sql

- LINQ ne olduğunu ve nasıl kullanıldığını öğrendim. LINQ kodlarım içerisinde, farklı dil kullanmadan, çeşitli ortamlardaki verileri sorgulayıp, filtrelemeyi sağladığımı öğrendim. LINQ sorgularının bir dizi işlev çağrısından oluşur ve bu işlevlerin, verileri filtreleme, sıralama, gruptama ve toplama gibi çeşitli işlemleri gerçekleştirebildiğini; LINQ sorgularının .NET Framework'ün bir dizi farklı veri kaynağında, örneğin diziler, nesneler, XML dosyaları ve veritabanlarında çalışabildiğini öğrendim. .Net'te sql sorguları yazmayı sağladığımı öğrendim.
- Sql'de rollback ile işlemlerin nasıl geri alındığını ve önemini öğrendim. Rollback; yapılan işlemleri geri aldığını, bir tablodaki verileri DELETE, UPDATE, INSERT gibi işlemler yapılırken ROLLBACK ile işlemin geri alınabildiğini öğrendim. Öncelikle begin transaction komutu ile işlemin başlayacağını belirttim. Insert into ve values komutları ile tabloya ekleme yaptım. Commit transaction komutu ile işlemlerin kalıcı olmasını sağladım. Begin catch komutundan sonra xact\_state komutu ile herhangi bir transaction var mı diye kontrol ettim. Sonrasında rollback transaction komutuyla yapılan işlemlerin geri alınmasını sağladım ve end catch komutuyla sonlandırdım.



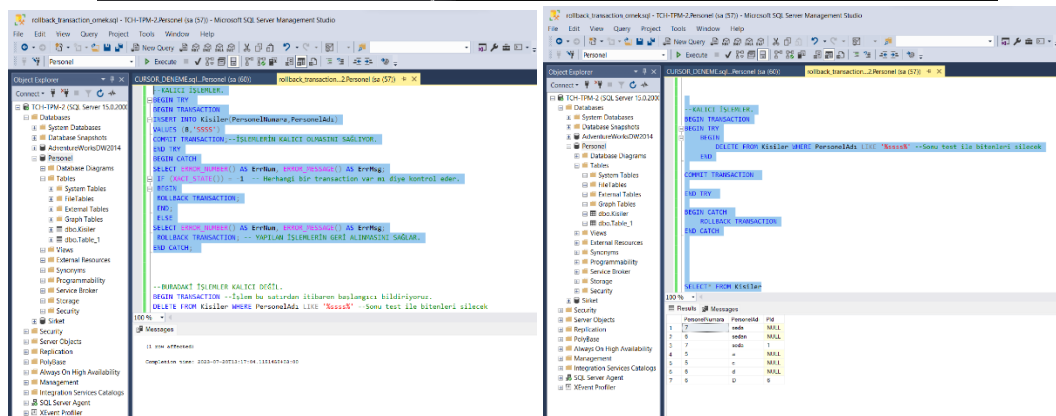
```
using System;
using System.Linq;

class Program
{
    static void Main(string[] args)
    {
        int[] numbers = new int[] { 23, 34, 56, 87, 85, 94, 99, 100 };

        IEnumerable<int> numb =
            from number in numbers
            where number > 70
            select number;

        foreach (var i in numb)
        {
            Console.WriteLine(i);
        }

        Console.ReadLine();
    }
}
```



```
-- ROLLBACK İZLEMLERİ
BEGIN TRANSACTION
INSERT INTO Kişiler (PersonelId, KişilerId)
VALUES (8, 1055)
COMMIT TRANSACTION --İŞLEMLERİN KALICI OLMASINI SAĞLIYOR
END TRY
BEGIN CATCH
SELECT @@TRANCOUNT AS ErrorNum, @@TRANCOUNT AS ErrorMsg;
IF @@TRANCOUNT < 1 -- 1. Herhangi bir transaction var mı diye kontrol eder.
RAISERROR (N'Transaction is not in a valid state.', 16, 1)
ROLLBACK TRANSACTION;
END
--BİRBAŞKA TİP ÖRNEK
BEGIN TRANSACTION --İşlem ile veritabanı kullanılarak kilitlenir.
DELETE FROM Kişiler WHERE PersonelId = 10; --Kişiler --Soru test ile bitirilecek
END CATCH
```

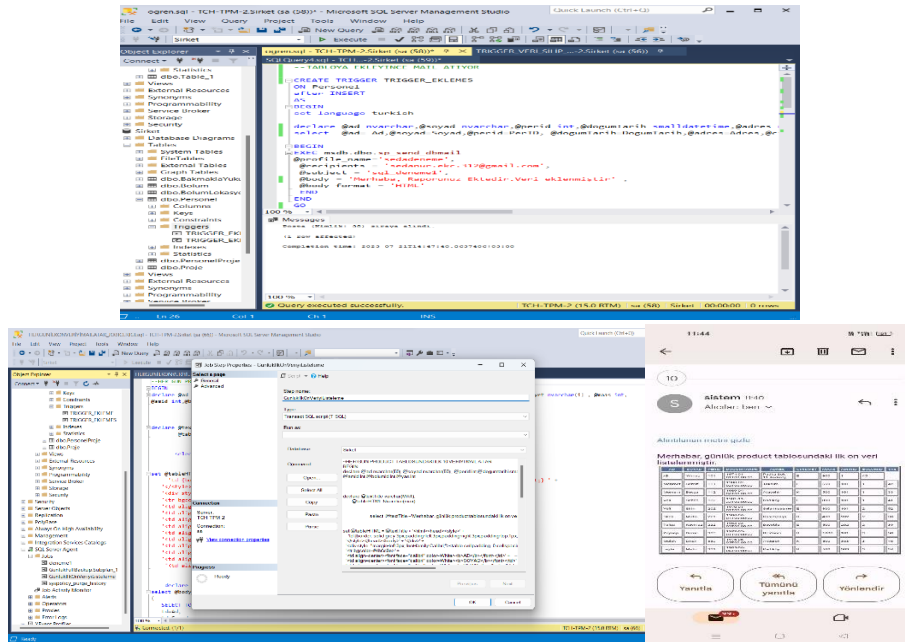
KONTROL SONUCU:

KISIM:

TARİH:21.07.2023

YAPILAN İŞ: Sql

- Sql ile ilgili birkaç tane görev verdiler ve bunları tamamladım. Bu verilen görevlerden ilki herhangi bir tabloya veri ekleyince triggers kullanarak mail atma, ikincisi tablodan veri silince triggers kullanarak mail atma sonuncusu ise her gün herhangi bir tablodaki ilk 10 veriyi mail atmak. İlk görev olan tabloya veri ekleyince triggers kullanarak mail atmayı öncelikle create trigger komutunu kullanarak trigger oluşturup on komutundan sonra trigger'a isim verdim. After insert komutuyla ekleme yaptıktan sonra işlem yapılacağını belirttim. Begin komutundan sonra declare komutuyla değişkenler tanımladım ve veri eklemek istediğim tablodaki sütun isimleriyle eşleştirdim. Tekrar begin komutuyla mail gönderme şablonu hazırladım ve end'leri yazarak bitirdim. Daha sonrasında tabloya veri ekleyerek mailin gönderildiğini test ettim. Bir diğer görev olan tablodan veri silince triggers kullanarak mail atmayı ise tabloya veri ekleme gibi yaptım ancak after insert komutu yerine after delete komutuyla tablodan veri silindikten sonra mail atmasını sağladım. Son görev olan her gün herhangi bir tablodaki ilk 10 veriyi mail atmayı ise job oluşturarak yaptım. Job oluşturduktan sonra steps kısmında öncelikle tablodaki değişkenleri declare komutu ile tanımladım. Sonrasında bir bilgilendirme mesajı yazdım ve sonrasında yazdığım sorgu ile birleştirerek tablodaki verileri ekledim. Son olarak ise mail gönderme şablonunu oluşturdum. Schedules kısmında her gün günde bir kere çalışma saatini belirledim. Daha sonrasında saatini düzenlediğimde mail gönderme işleminin başarılı oldu.



KONTROL SONUCU:

Kocaeli Üniversitesi Mühendislik Fakültesi