

SQL- Yapısal Sorgulama Dili

Adem AKKUŞ

| Bilgisayar Mühendisi | Uzm. Bilişim Tekn. Öğrt. | Eğitimci

Adem AKKUŞ

| Bilgisayar Mühendisi | Uzm. Bilişim Tekn. Öğrt. | Eğitimci

Veri Tanımlama Dili (DDL-Data Definition Language)

- Veri tanımlama dili verinin ne olduğu ile değil verinin nerede ve nasıl tutulacağı ile ilgilenir.
- Veri tanımlama dilinin temel ifadeleri
 - **Create**; Nesne oluşturmak için kullanılır
 - **Alter**; Nesneler üzerinde değişiklik yapmak için kullanılır.
 - **Drop**; Nesneleri silmek için kullanılır.

Veri Tanımlama Dili - Create

- Create ; Veritabanı nesnelerini ve veritabanının kendisi oluşturmak için kullanılır.
 - CREATE DATABASE veritabanı_ismi
 - CREATE TABLE tablo_ismi ...
 - CREATE FUNCTION ...
 - CREATE PROCEDURE ...
 - ...

Veri Tanımlama Dili – Create Database

```
CREATE DATABASE db_dersOrnek
```

```
ON
```

(NAME = db_dersOrnek	--(Dosya Adı)
, FILENAME= 'c:\db_dersOrnek.mdf'	--(Fiziksel Dosya Adı)
, SIZE = 10	--(Dosya Boyutu)
, MAXSIZE = 50 -	-(Maksimum Dosya Boyutu)
, FILEGROWTH =5)	--(Artım Miktarı)

```
LOG
```

```
ON
```

```
(NAME = db_dersOrnek  
, FILENAME= 'c:\db_dersOrnek.ldf'  
, SIZE = 5  
, MAXSIZE = 40  
, FILEGROWTH =2)
```

Veri Tanımlama Dili – Create Table

- Veritabanı üzerinde yeni tablo oluşturur.
 - Sütun adları
 - Veri tipleri
 - Kısıtlamalar (Constraintler)
 - Otomatik arttırma
 - NULL yada Not NULL
- Kullanım Şekli

CREATE TABLE <tablo_adı> (alan_adı
veri_türü)

Veri Tanımlama Dili – Create Table

```
CREATE TABLE tbl_ogrenci
```

```
(
```

```
  ogr_id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
```

```
  ogrNo NVARCHAR(8) NOT NULL,
```

```
  tcNo DECIMAL(11) NOT NULL UNIQUE,
```

```
  ad NVARCHAR(100) NOT NULL,
```

```
  soyad NVARCHAR(100) NOT NULL,
```

```
  dogumTarihi DATETIME NULL,
```

```
  dogumYeri NVARCHAR(50)
```

```
)
```

Veri Tanımlama Dili – Create Table

```
CREATE TABLE tblOgrenci(  
    id INT IDENTITY(1,1) NOT NULL,      --boş bırakılamaz  
    adi NVARCHAR(25),  
    soyadi NVARCHAR(50),  
    notu TINYINT,  
    kayittarihi DATETIME  
)
```

Veri Tanımlama Dili – Create Table

CREATE TABLE

tbl_ogrenciNot (

ogrNot_id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

ogr_id INT NOT NULL FOREIGN KEY REFERENCES tbl_ogrenci(ogr_id),

ders_id INT NOT NULL FOREIGN KEY REFERENCES

tbl_ders(ders_id), notu TINYINT CHECK (notu>=0 and notu<=100),

notGirisTarihi DATETIME NOT NULL DEFAULT GETDATE()

)

Veri Tanımlama Dili – Create Table

```
CREATE TABLE tbl_ogrenciNot(  
    ogrNot_id INT IDENTITY(1,1) NOT NULL,  
    ogr_id INT NOT NULL,  
    ders_id INT NOT NULL,  
    notu TINYINT,  
    notGirisTarihi DATETIME NOT NULL  
    CONSTRAINT pk_key PRIMARY KEY (ogrNot_id),  
    CONSTRAINT ck_not CHECK (notu>=0 AND notu <=100),  
    CONSTRAINT uk_ogr_id_ders_id UNIQUE (ogr_id,ders_id),  
    CONSTRAINT fk_key_ogr_id FOREIGN KEY (ogr_id) REFERENCES  
tbl_ogrenci(ogr_id),  
    CONSTRAINT fk_key_ders_id FOREIGN KEY (ders_id) REFERENCES  
tbl_ders(ders_id) ON UPDATE CASCADE ON DELETE CASCADE  
)
```

Veri Tanımlama Dili - ALTER

- Alter; Daha önce oluşturulmuş veritabanı nesnesinin özelliklerini değiştirmek için kullanılır.
 - **ALTER TABLE** *tablo ismi* ...
 - ALTER FUNCTION ...
 - ALTER PROCEDURE ...
 - *ALTER TRIGGER*
 - ...

Veri Tanımlama Dili - ALTER

- ALTER TABLE tablo ADD sütun_adı özellikler

ALTER TABLE ogrenci

ADD dogumTarihi DateTime NOT NULL

- ALTER TABLE tablo DROP COLUMN sütun_adı

ALTER TABLE ogrenci

DROP COLUMN dogumTarihi

- ALTER TABLE tablo ALTER COLUMN
sütun_adı özellikler

ALTER TABLE ogrenci

ALTER COLUMN dogumTarihi DateTime NULL

Veri Tanımlama Dili - ALTER

- ALTER TABLE ogrenci ADD UNIQUE (ogrNo)
- ALTER TABLE ogrenciNot ADD UNIQUE (ders_id,ogr_id)
- ALTER TABLE ogrenci ADD PRIMARY KEY (ogr_id)
- ALTER TABLE ogrenci ADD CONSTRAINT pk PRIMARY KEY (ogr_id)
- ALTER TABLE ogrenciNot ADD CONSTRAINT fk FOREIGN KEY (ogr_id) REFERENCES ogrenci(ogr_id)
- ALTER TABLE ogrenciNot ADD CHECK (notu>=0 and notu <=100)

Veri Tanımlama Dili - DROP

- Drop; Veritabanı nesnelerini ve veritabanının kendisi silmek için kullanılır.
 - DROP DATABASE veritabanı ismi
 - DROP TABLE tablo ismi ...
 - DROP FUNCTION ...
 - DROP PROCEDURE ...
 - ...

Veri Tanımlama Dili - DROP


- DROP TABLE öğrenci
- DROP DATABASE okul

Veri Tanımlama Dili - DROP

```
8 CREATE DATABASE SilinecekDB
9 USE SilinecekDB
10 DROP DATABASE SilinecekDB
```

Messages

Msg 3702, Level 16, State 3, Line 10
Cannot drop database "SilinecekDB" because it is currently in use.



Completion time: 2024-12-08T10:14:09.8682824+03:00

Veri Tanımlama Dili - DROP

```
8 CREATE DATABASE SilinecekDB
9 USE SilinecekDB
10 USE master
11 DROP DATABASE SilinecekDB
```

259 %

Messages

Commands completed successfully.

Completion time: 2024-12-08T10:16:44.6586517+03:00

Veri Tanımlama Dili -TRUNCATE TABLE

SQL ,**TRUNCATE TABLE** komutu, tablonun yapısını yeniden başlatarak mevcut bir tablodaki tüm kayıtları silmek için kullanılır.

Bu komut, veritabanına bir tablodaki tüm kayıtlar için alanı serbest bırakması ve tablo boyutunu sıfırlayarak bu tablonun yapısını değiştirmesi talimatını verir.

Bu nedenle, tüm tablo verileri kaldırılrsa bile Veri Manipülasyon Dili (DML) yerine Veri Tanımlama Dili (DDL) işlemi olarak kabul edilir.

Veri Tanımlama Dili -TRUNCATE TABLE

Mantıksal olarak, **TRUNCATE TABLE** deyimi, DELETE TABLE deyimine benzer şekilde çalışır, ancak WHERE yan tümcesi yoktur. Ancak TRUNCATE, DELETE'den çok daha hızlıdır ve işlendikten sonra geri almaya izin vermez.

Bir tabloyu silmek için DROP TABLE komutunu da kullanabilirsiniz, ancak bu, tüm tablo yapısını veritabanından kaldıracaktır ve bazı verileri yeniden depolamak istiyorsanız bu tabloyu yeniden oluşturmanız gerekecektir.

Veri Tanımlama Dili -TRUNCATE TABLE

- TRUNCATE TABLE <tablo ismi>
- TRUNCATE TABLE CUSTOMERS
- TRUNCATE TABLE tblOgrenci

DELETE vs TRUNCATE

TRUNCATE ve DELETE komutları mantıksal olarak benzer şekilde çalışsa da, aralarında bazı önemli farklılıklar vardır.

DELETE

TRUNCATE

SQL'deki DELETE komutu, WHERE Yan tmcesinde belirtilen kořullara baęlı olarak bir tablodan bir veya daha fazla satırı kaldırır.	SQL'in TRUNCATE komutu, herhangi bir kořulun karřılanıp karřılanmadıęına bakılmaksızın bir tablodaki tm satırları kaldırmak iin kullanılır.
Bu bir DML(Veri Maniplasyon Dili) komutudur.	Bu bir DDL(Veri Tanımlama Dili) komutudur.
DELETE komutunda deęiřiklik yapıldıktan sonra deęiřikliklerin commit edilmesi iin manual COMMIT yapılmasına ihtiya vardır.	TRUNCATE komutunu kullandıęınızda tabloda yapılan deęiřiklikler otomatik olarak taahht edilir.
Satırları teker teker siler ve her silme iřlemi iin bazı ltler uygular.	Tek seferde tm bilgileri kaldırır.
WHERE yan tmcesi bu durumda kořul olarak iřlev grr.	WHERE Cmlesi kullanmaya gerek yoktur.
Silme iřleminden sonra tm satırlar kilitletir.	TRUNCATE, sayfaları silinemeyecek řekilde kilitleyen bir tablo kilidi kullanır.
Gnlk dosyasındaki her iřlemin kaydını yapar.	Kaydedilen tek etkinlik, verilerin depolandıęı sayfaların serbest bırakılmasıdır.
TRUNCATE komutuna kıyasla daha fazla iřlem alanı tketir.	Nispeten daha az iřlem alanı gerektirir.
Bir kimlik stunu varsa tablo kimlięi, tablo oluřturulduęunda sahip olduęu deęere sıfırlanmaz.	Tablo kimlięini, tohum olarak verilen bir deęere dndrr.
Silmek iin yetki gerektirir.	Tablo deęiřtirme izni gerektirir.
Byk veritabanları sz konusu olduęunda, ok daha yavařtır.	Bu daha hızlı.

Adem AKKUř

| Bilgisayar Mhendisi | Uzm. Biliřim Tekn. ęrt. | Eęitmen

DROP vs TRUNCATE

Tablo yapısını sıfırlayan TRUNCATE'in aksine, DROP komutu tablo alanını bellekten tamamen serbest bırakır. Veritabanı nesnelerinin tanımlarıyla etkileşime girdiklerinde her ikisi de Veri Tanımlama Dili işlemleridir; bu, veritabanının bu komutlar yürütüldükten sonra geri alma şansı olmadan otomatik olarak taahhütte bulunmasına olanak tanır.

DROP

TRUNCATE

SQL'deki DROP komutu, tanımı, izinleri, kısıtlamaları, verileri vb. dahil olmak üzere tüm bir tabloyu bir veritabanından kaldırır.	TRUNCATE komutu, herhangi bir koşulun sağlanıp sağlanmadığına bakılmaksızın tablodaki tüm satırları kaldırmak için kullanılır ve tablo tanımını sıfırlar.
Bu bir DDL(Veri Tanımlama Dili) komutudur.	Aynı zamanda bir DDL(Veri Tanımlama Dili) komutudur.
Tablo alanı bellekten tamamen boşaltılır.	Tablo hala bellekte var.
Tüm bütünlük kısıtlamaları kaldırılır.	Bütünlük kısıtlamaları tabloda hala mevcuttur.
Bu komutu gerçekleştirebilmek için sırasıyla tablo şeması ve tablo üzerinde ALTER ve CONTROL izinlerini gerektirir.	Yalnızca tabloyu kesmek için ALTER izinlerini gerektirir.
DROP komutu TRUNCATE komutundan çok daha yavaş, DELETE komutundan ise daha hızlıdır.	Hem DROP hem de DELETE komutlarından daha hızlıdır.

Veri İşleme Dili

(DML-Data Manipulation Language)

- Veri işleme dili tutulan veriler üzerinde işlem yapar.
- Veri işleme dilinin temel ifadeleri
 - **Select**; Veritabanındaki kayıtları sorgular
 - **Insert**; Yeni kayıt eklemek için kullanılır.
 - **Update**; Daha önce olan kaydı günceller
 - **Delete**; Veritabanındaki kaydı silmek için kullanılır.

Veri İşleme Dili - SELECT

- Select; Tablo yada tablolarda istenilen verinin seçimi için kullanılır. Seçilen verilerde değişim olmaz.
 - **SELECT** sütunlar **FROM** tablo ismi
 - **select** * **from** ogrenciler
 - **select** ogrNo, ad, soyad **from** ogrenciler

Veri İşleme Dili - SELECT

Özellikleri

- Tablonun her sütunu yazılabilir.
- Birden fazla tabloda sorgulama yapılabilir.
- Mevcut kayıtlar belirli koşullara göre seçilebilir.
- Sorgulama sırasında birden fazla sütun birleştirilebilir veya sütunlara ek bilgiler eklenebilir.
- Sorgulamada sütunlar sayısal tipte ise matematiksel işlemler yapılabilir.
- Sorgulamada dönen değerler birden fazla sütuna göre sıralı seçilebilir.
- Ortak sütunlara sahip tablolar ilişkilendirilerek farklı tablolardan ortak özelliklere sahip sütunlar seçilebilir.
- Daha karmaşık sorgular için iç içe select yazılabilir.

Veri İşleme Dili - SELECT

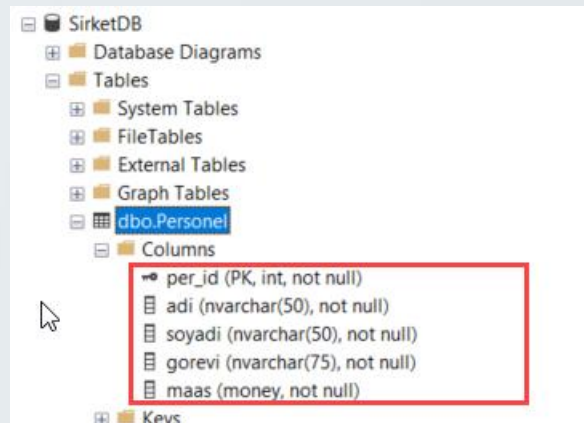
Personel Tablosu

per_id	adi	soyadi	gorevi	sehir	maas
1	Ali	Coşkun	Mühendis	Denizli	10000
2	Mehmet	Kara	Tekniker	Ankara	20000
3	Ayşe	Con	Mühendis	Antalya	30000

- `Select * from tbl_Personel`
- `select per_id, adi, gorevi from personel`

Veri İşleme Dili - SELECT

DESKTOP-L2N63DT....DB - dbo.Personel		
Column Name	Data Type	Allow Nulls
per_id	int	<input type="checkbox"/>
adi	nvarchar(50)	<input type="checkbox"/>
soyadi	nvarchar(50)	<input type="checkbox"/>
gorevi	nvarchar(75)	<input type="checkbox"/>
maas	money	<input type="checkbox"/>
		<input type="checkbox"/>



Veri İşleme Dili - SELECT

- Sorgulamada belirli koşullara bağlı kayıtlar isteniyorsa **Where** ifadesi kullanılır.

- <, >, <=, >=, =, <>, !=
- AND, OR, NOT
- IN
- BETWEEN
- LIKE

Select * From tbl_personel where adi='Ali'

Personel Tablosu

per_id	adi	soyadi	gorevi	sehir	maas
1	Ali	Coşkun	Mühendis	Denizli	10000
2	Mehmet	Kara	Tekniker	Ankara	20000
3	Ayşe	Con	Mühendis	Antalya	30000

Veri İşleme Dili - SELECT

Personel Tablosu

per_id	adi	soyadi	gorevi	sehir	maas
1	Ali	Coşkun	Mühendis	Denizli	10000
2	Mehmet	Kara	Tekniker	Ankara	20000
3	Ayşe	Con	Mühendis	Antalya	30000

- SELECT * FROM tbl_personel WHERE per_id<3
- SELECT * FROM tbl_personel WHERE gorevi='Mühendis' AND sehir='Denizli'
- SELECT * FROM tbl_personel WHERE sehir='Denizli' OR soyadi='Con'
- SELECT * FROM tbl_personel WHERE sehir IN ('Denizli','Ankara')
- SELECT * FROM tbl_personel WHERE ad LIKE 'A%'

Veri İşleme Dili - INSERT

- Insert; Tabloya yeni veri girişi için kullanılır.
- Kullanım şekilleri
 - **INSERT INTO** tablo_ismi **VALUES** (değerler)
 - **insert into** tbl_personel **values**
(‘Kemal’,‘Kara’,‘Güvenlik’,‘Denizli’)
 - **INSERT INTO** tablo_ismi (sutunAdları)
VALUES (değerler)
 - **insert into** tbl_personel (adi,soyadi,gorevi,sehir)
values (‘Kemal’,‘Kara’,‘Güvenlik’,‘Denizli’)

Veri İşleme Dili - INSERT

- Insert; Tabloya yeni veri girişi için kullanılır.
- Kullanım şekilleri

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');

Veri İşleme Dili - INSERT

- Yalnızca belirtilen sütunlara yeni veri girişi için kullanılabilir.
- Kullanım şekilleri

INSERT INTO Customers (CustomerName, City, Country)

VALUES ('Cardinal', 'Stavanger', 'Norway');

Veri İşleme Dili - INSERT

- Birden çok veri satırı eklemek için kullanılabilir.
- Kullanım şekilleri

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)

VALUES

('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway'),

('Greasy Burger', 'Per Olsen', 'Gateveien 15', 'Sandnes', '4306', 'Norway'),

('Tasty Tee', 'Finn Egan', 'Streetroad 19B', 'Liverpool', 'L1 0AA', 'UK'));

Veri İşleme Dili - UPDATE

- Update; Tabloya daha önce girilmiş verilerin güncellenmesi için kullanılır.

- **UPDATE** tablo_ismi
SET sütun1=deger...
WHERE koşul

```
UPDATE tbl_ogrenci  
SET tcNo=2  
WHERE tcNo=1
```

--TcNo sisteme yanlışlıkla 1 olarak kayıt
-- edilmiş 2 olması gerekiyor. tcNo su 1
-- olanın tcNo su 2 olarak güncellendi.

```
UPDATE tbl_ogrenciNot  
SET notu=notu+10  
WHERE ders_id=1
```

-- ders_id 1 olan hoca bu dersi alan herkese
-- +10 puan veriyor herkesin notunu
-- 10 puan arttırıp güncelledik

```
4 UPDATE Personel
5 SET gorevi='İnşaat Mühendisi'
6 WHERE per_id=2
7 --Elif GÖKMEN görevi Endüstri Mühendisi olarak değiştir.
8 UPDATE Personel
9 SET gorevi='Endüstri Mühendisi'
10 WHERE adi='Elif' AND soyadi='GÖKMEN'
11 --WHERE per_id=3
12 --Burhan ADEM maaş güncelle
13 UPDATE Personel
14 SET maas=60000
15 WHERE per_id=2
16 --maaşı 0 olanlara asgari ücret ataması yapalım
17 UPDATE Personel
18 SET maas=20000
19 WHERE maas=0
20 ----maaşı 30000 az olanlara zam yapalım
21 UPDATE Personel
22 SET maas=maas+15000
23 WHERE maas<30000
```

Veri İşleme Dili - DELETE

- Delete; Tabloya daha önce girilmiş verilerin silinmesi için kullanılır.

– **DELETE FROM** tablo_ismi
WHERE koşul

DELETE FROM tbl_ders
WHERE dersKod ='geng 103'

--bu ders yanlışlıkla eklenmiş
--siliyoruz.

DELETE FROM tbl_ders
WHERE ders_id=1

--bu dersi silmeye çalışalım

Veri İşleme Dili - DELETE

Query3.sql - (L2N63DT\mypc (52))*

```
1 USE SirketDB
2 GO
3 --DELETE FROM tablo_ismi
4 ---WHERE şart
5 DELETE FROM Personel
6 WHERE per_id=8
7 ---
8 DELETE FROM Personel
9 WHERE soyadi='ATEŞ' AND adi='Can'
10
```

DESKTOP-L2N63DT....DB - dbo.Personel

	per_id	adi	soyadi	qorevi	maas
▶	1	Adem	AKKUŞ	Bilgisayar Mühendisi	50000,0000
	2	Burhan	Adem	İnşaat Mühendisi	60000,0000
	3	Elif	GÖKMEN	Endüstri Mühendisi	35000,0000
	5	Berk	BULUT	Bilgisayar Mühendisi	40000,0000
	9	Kerem	ALDEMİR	Yazılım Mühendisi	45000,0000
	12	Ceyda	ATEŞ	dsfsdsdfs	44444,0000
*	NULL	NULL	NULL	NULL	NULL

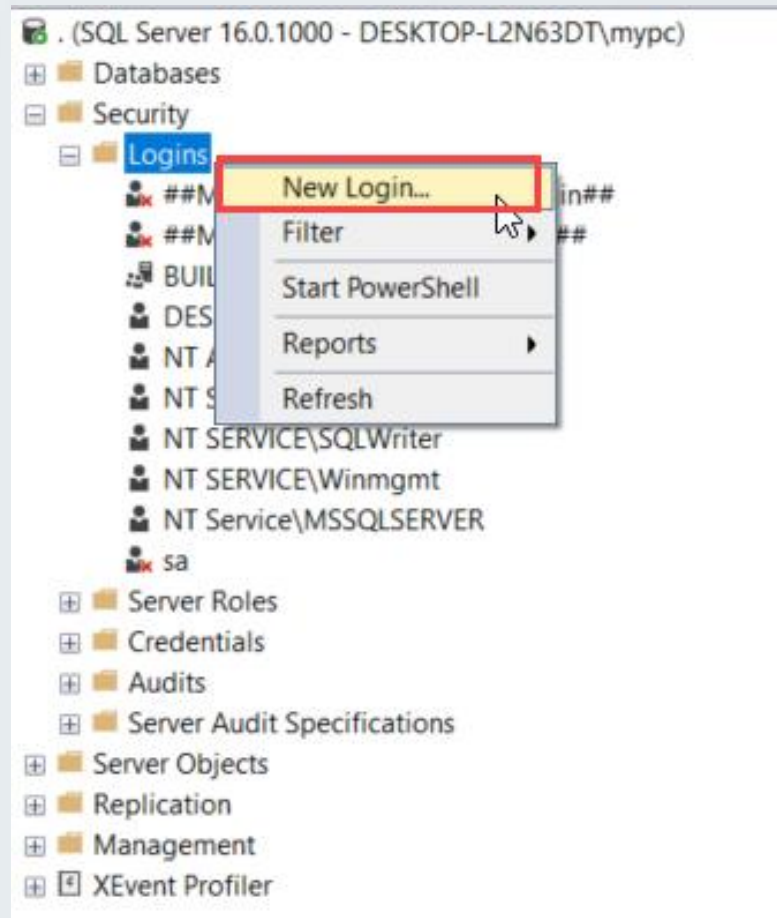
Messages

(1 row affected)

Veri Kontrol Dili (DCL-Data Control Language)

- Veri tabanı üzerindeki verilere erişim yetkilerini düzenler.
- Veri kontrol dilinin temel ifadeleri
 - **Grant**; Kullanıcıların kayıtlar üzerinde işlem yapmasına izin verir.
 - **Deny**; Kullanıcının kayıtları kullanmasını kısıtlar
 - **Revoke**; Daha önce yapılan tüm kısıtlama ve izinleri iptal eder.

Veri Kontrol Dili (DCL-Data Control Language)



Veri Kontrol Dili (DCL-Data Control Language)

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script ? Help

Login name: Search...

☐ Windows authentication

☐ Microsoft Entra ID authentication

☒ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☒ Enforce password policy

☒ Enforce password expiration

☒ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Add

Mapped Credentials

Credential	Prov
------------	------

Remove

Default database:

Default language:

Progress

Ready

OK Cancel

Veri Kontrol Dili (DCL-Data Control Language)

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script ? Help

Login name: Search...

☒ Windows authentication

☐ Microsoft Entra ID authentication

☐ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☒ Enforce password policy

☒ Enforce password expiration

☒ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Add

Mapped Credentials

Credential	Provider
------------	----------

Remove

Default database:

Default language:

Progress

Ready

OK Cancel

Veri Kontrol Dili - GRANT

- Grant; Kullanıcıya veritabanı veya nesneleri üzerinde çeşitli izinler vermek için kullanılır.

- **GRANT** izinler
- **ON** izin alanı
- **TO** kullanıcılar

Veritabanı için	Tablo İçin	Görünüm İçin
CREATE DATABASE BACKUP DATABASE CREATE TABLE CREATE VIEW	DELETE INSERT SELECT UPDATE	DELETE INSERT SELECT UPDATE

GRANT DELETE,INSERT, SELECT,UPDATE

ON tbl_personel

TO dersKull

Veri Kontrol Dili - DENY

- Deny; Kullanıcıya veritabanı veya nesneleri üzerinde çeşitli izinleri kısıtlamak için kullanılır.
 - **DENY** izinler
 - **TO** kullanıcılar

VeriTabanı için	Tablo İçin	Görünüm İçin
CREATE DATABASE BACKUP DATABASE CREATE TABLE CREATE VIEW	DELETE INSERT SELECT UPDATE	DELETE INSERT SELECT UPDATE

DENY DELETE,INSERT, SELECT,UPDATE TO dersKull

Veri Kontrol Dili - REVOKE

- Revoke; Kullanıcıya veritabanı veya nesneleri üzerinde verilen izinleri ve kısıtlamalara kaldırmak için kullanılır.

- REVOKE izinler
- TO kullanıcılar

VeriTabanı İçin	Tablo İçin	Görünüm İçin
CREATE DATABASE BACKUP DATABASE CREATE TABLE CREATE VIEW	DELETE INSERT SELECT UPDATE	DELETE INSERT SELECT UPDATE

REVOKE DELETE, INSERT, SELECT, UPDATE TO dersKull

Örnek Veritabanı Tasarımı 2

Basit Bir Örnek

Adem AKKUŞ

| Bilgisayar Mühendisi | Uzm. Bilişim Tekn. Öğrt. | Eğitimci

Örnek Veritabanı Tasarımı -2

- Hazırlayacağımız veritabanı bir şirketin
 - aldığı ve sattığı ürün bilgilerini
 - ürünleri hangi firmalardan ne zaman ve ne kadar alındığını
 - hangi firmalara ne zaman ve kadar sattığı bilgilerini tutacaktır.
- İlk iş kavramsal tasarımı yapmak.

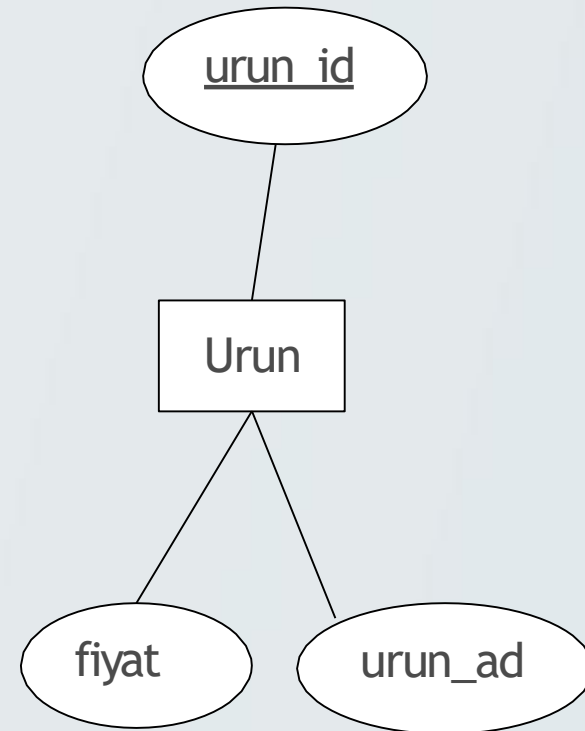
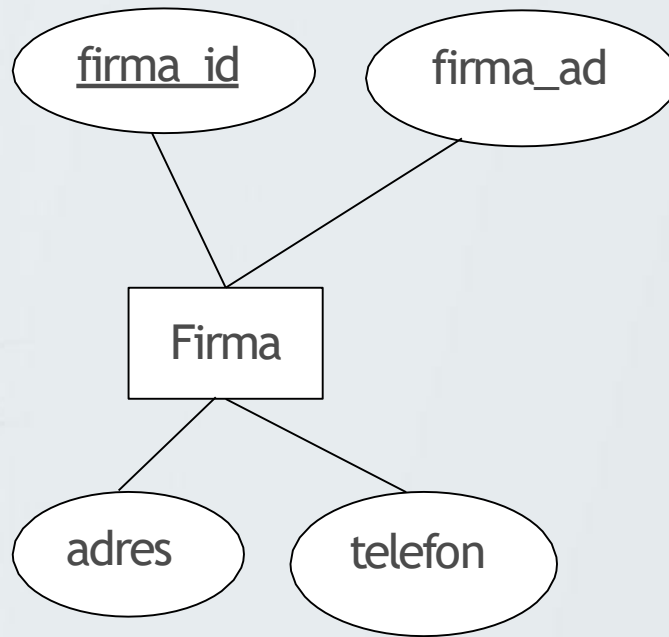
Örnek Veritabanı Tasarımı -2

- Varlıklarımız
 - Ürünler, Firmalar
- Niteliklerimiz
 - Urunler
 - urun_id, urunAd, fiyat
 - Firmalar
 - firma_id, firmaAd, telefon, adres

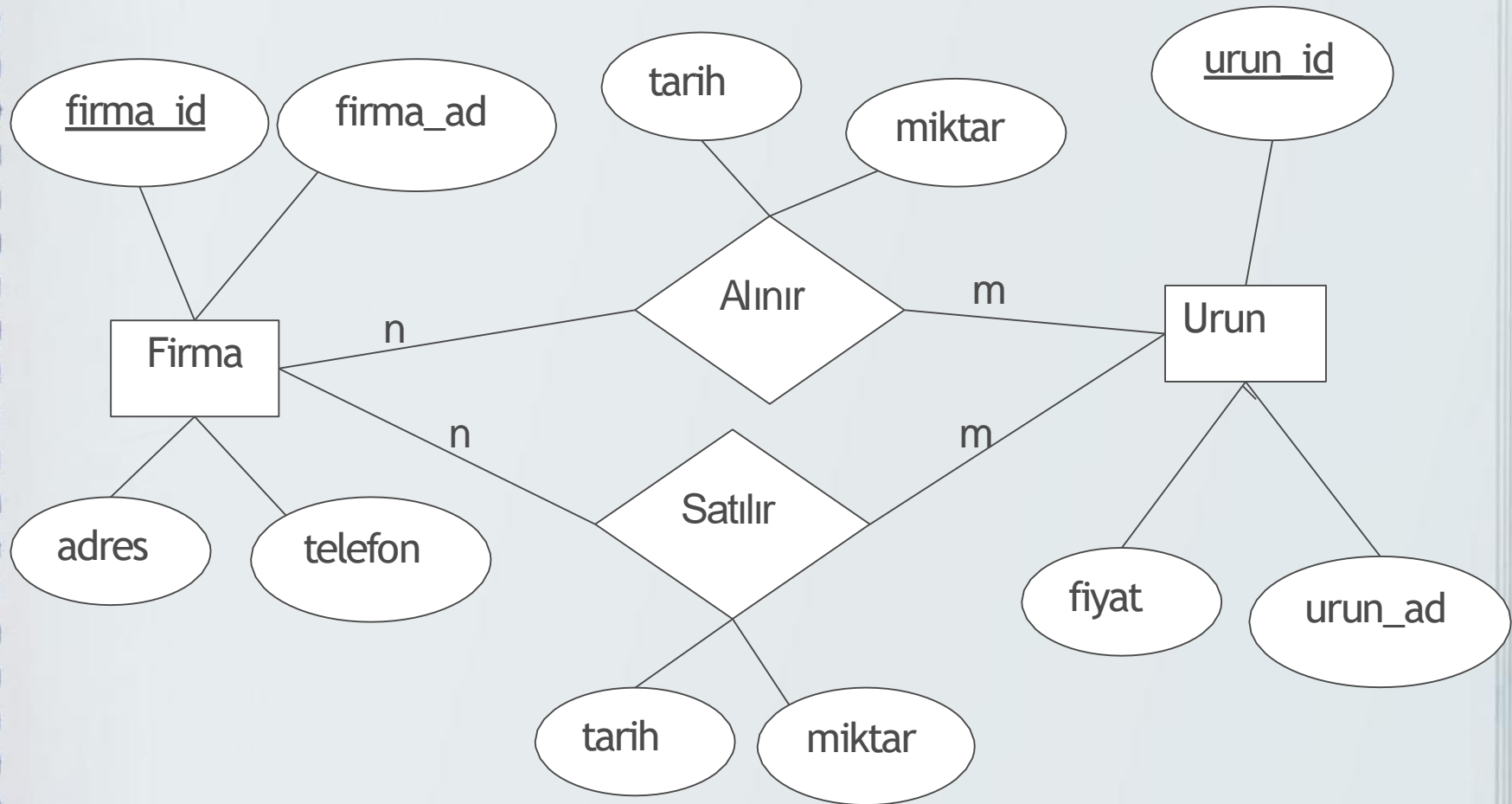
Kullanılan Semboller

Sembol	Açıklama
	Varlık kümesi
	Nitelik
	Anahtar nitelik
	İlişki
	Çok değerli nitelik
	Zayıf varlık kümesi

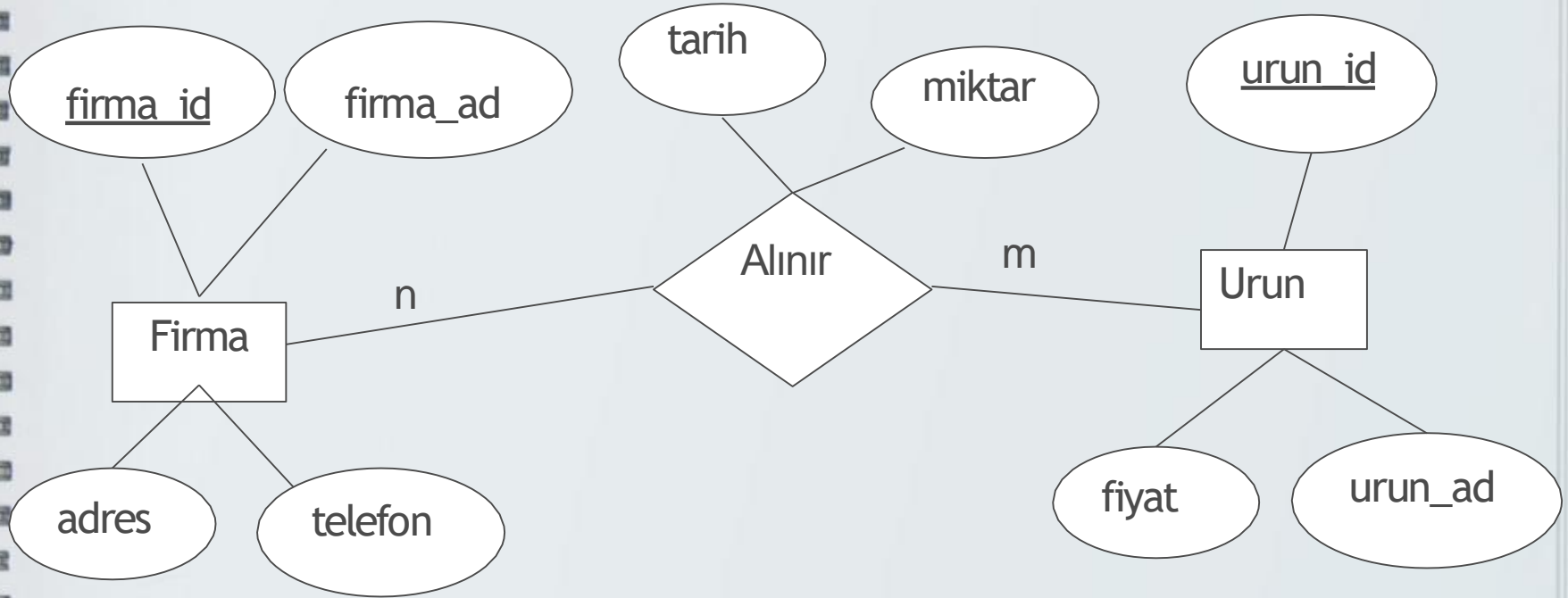
Varlıklar ve Nitelikleri



Kavramsal Tasarım

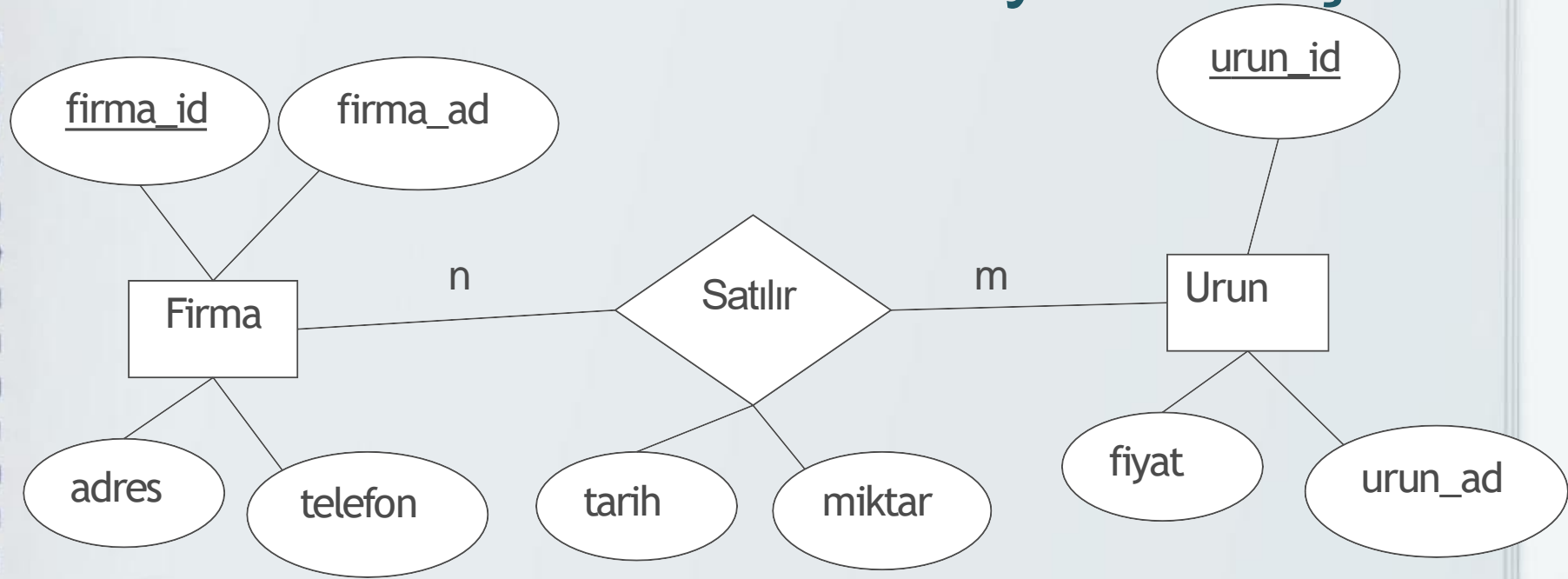


Kavramsal Tasarımın Tabloya Dönüşümü



- Firma (**firma_id**, firma_ad, adres, telefon)
- Urun (**urun_id**, urun_ad, fiyat)
- AlinanUrun (tarih, miktar, **firma_id**, **urun_id**, **alinanUrun_id**)

Kavramsal Tasarımın Tabloya Dönüşümü

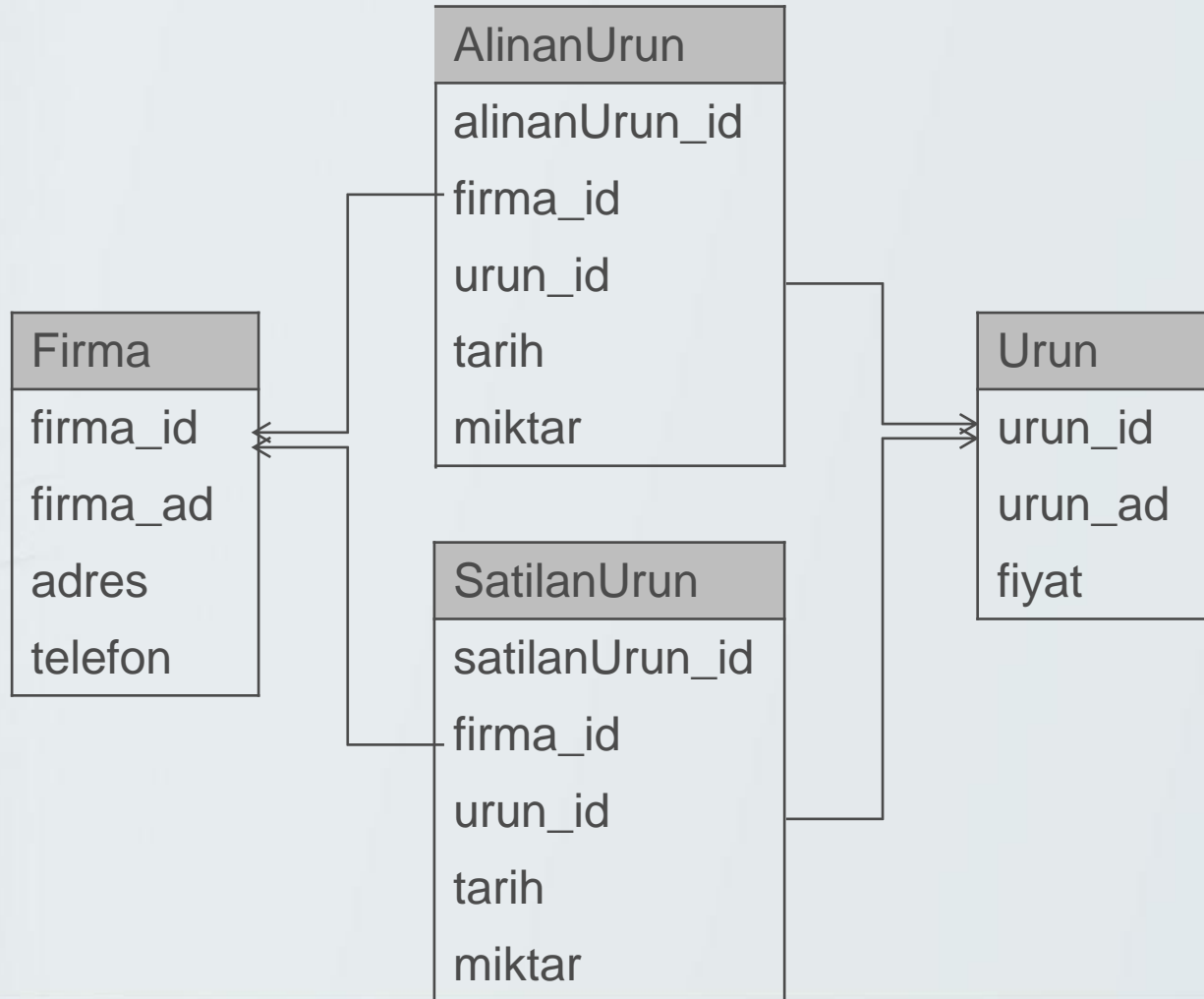


- Firma (firma_id, firma_ad, adres, telefon)
- Urun (urun_id, urun_ad, fiyat)
- SatılanUrun (tarih, miktar, firma_id, urun_id, satılanUrun_id)

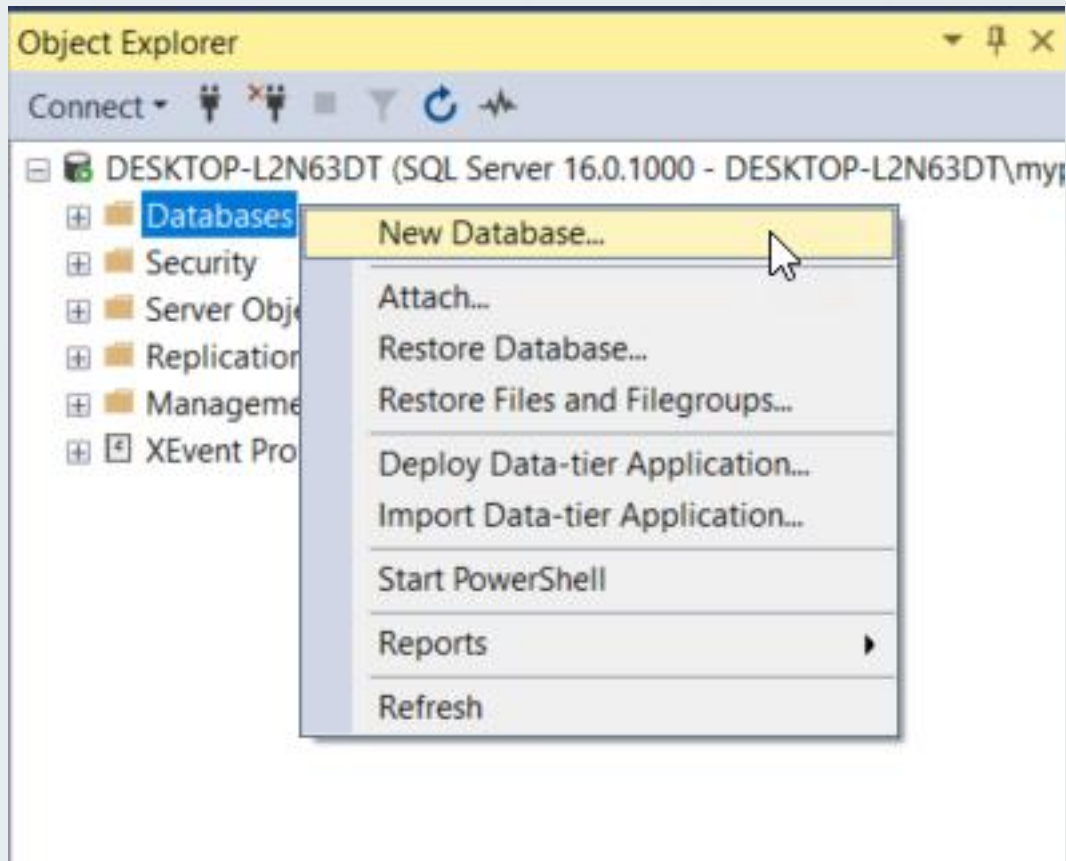
Kavramsal Tasarımın Tabloya Dönüşümü

- Firma (firma_id, firma_ad, adres, telefon)
- Urun (urun_id, urun_ad, fiyat)
- AlinanUrun (tarih, miktar, firma_id, urun_id, alinanUrun_id)
- SatilanUrun (tarih, miktar, firma_id, urun_id, satilanUrun_id)

Tablolar Arası İlişkiler



Tablolar Arası İlişkiler



Tablolar Arası İlişkiler

New Database

Select a page
General
Options
Filegroups

Script Help

MarketDB

Database name: MarketDB

Owner: <default>

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Path
MarketDB	ROWS ...	PRIMARY	8	By 64 MB, Unlimited	C:\Program
MarketDB_log	LOG	Not Applicable	8	By 64 MB, Unlimited	C:\Program

Connection

Server:
DESKTOP-L2N63DT

Connection:
DESKTOP-L2N63DT\mytc

[View connection properties](#)

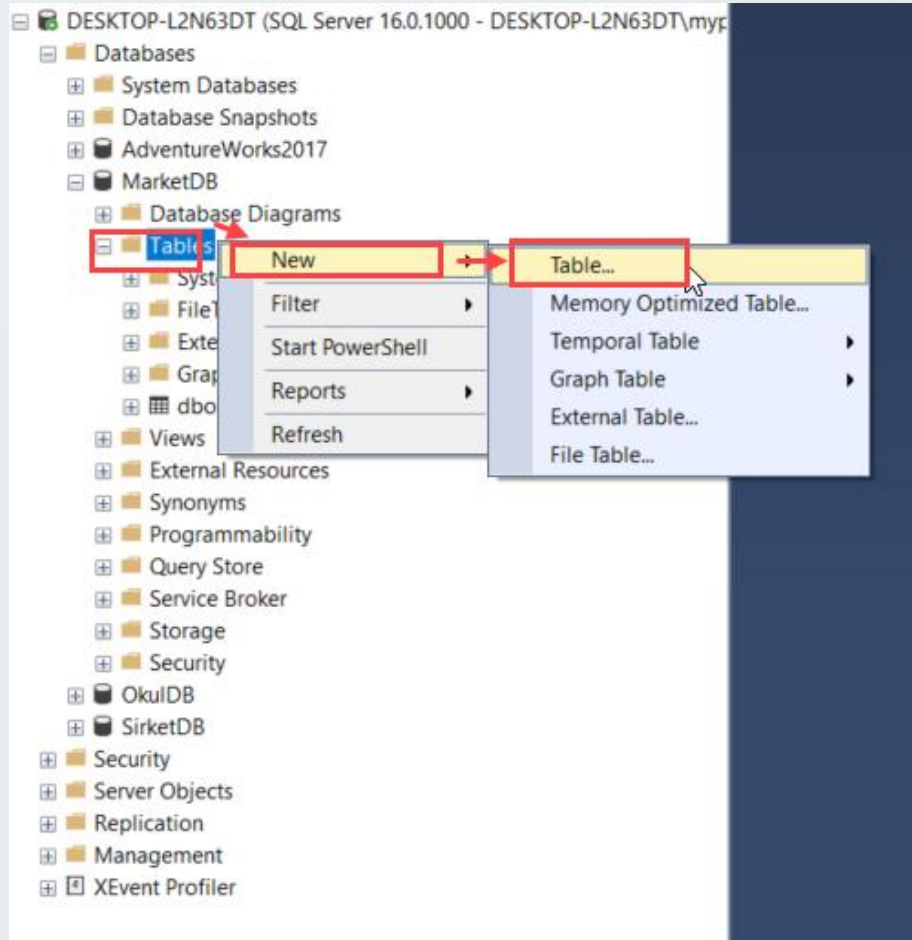
Progress

Ready

Add Remove

OK Cancel

Tablolar Arası İlişkiler



Tablolar Arası İlişkiler

Column Name	Data Type	Allow Nulls
firma_id	int	<input type="checkbox"/>

Column Properties	
(Name)	firma_id
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4

Column Name	Data Type	Allow Nulls
firma_id	int	<input type="checkbox"/>
firma_ad	nvarchar(100)	<input type="checkbox"/>
adres	nvarchar(150)	<input type="checkbox"/>
telefon	nvarchar(15)	<input checked="" type="checkbox"/>

Choose Name

Enter a name for the table:

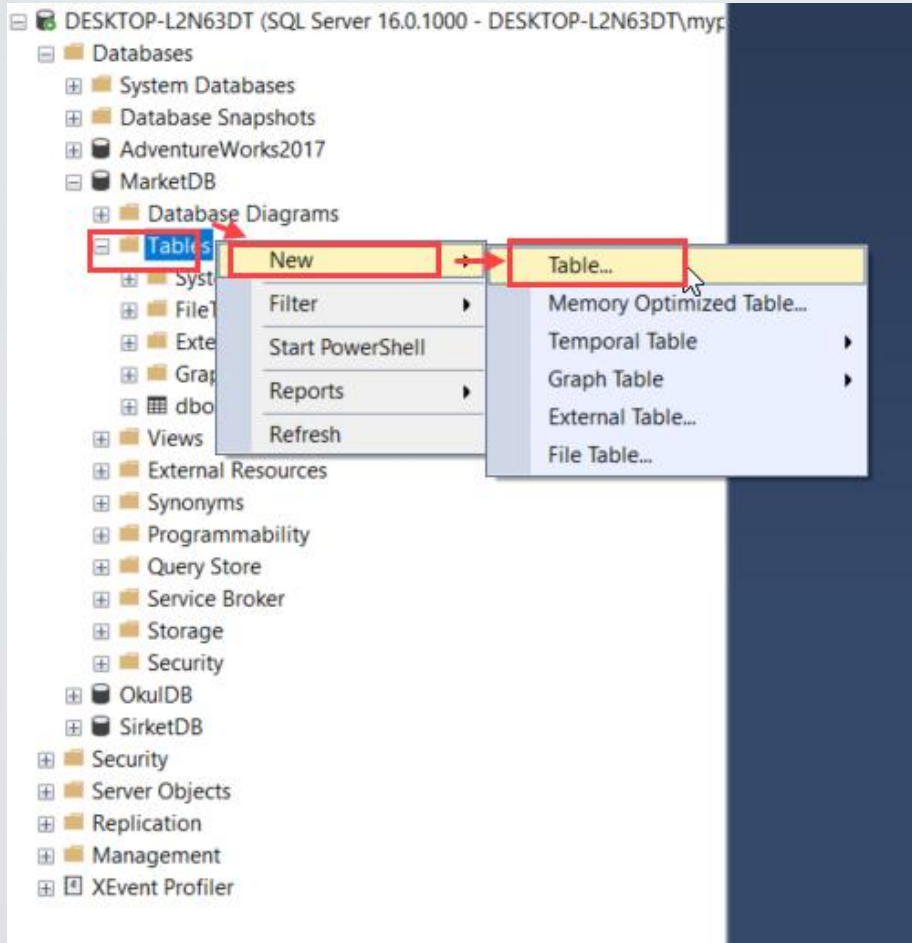
Firma

OK Cancel

Column Name	Data Type
firma_id	int
firma_ad	
adres	
telefon	

Set Primary Key

Tablolar Arası İlişkiler



Tablolar Arası İlişkiler

urun_id	int	<input type="checkbox"/>
		<input type="checkbox"/>

Column Properties	
(General)	
(Name)	urun_id
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No

Column Name	Data Type	Allow Nulls
urun_id	int	<input type="checkbox"/>
		<input type="checkbox"/>

Set Primary Key
Insert Column
Delete Column
Relationships...
Indexes/Keys...
Fulltext Index...
XML Indexes...
Check Constraints...
Spatial Indexes...
Generate Change Script...
Properties

Column Name	Data Type	Allow Nulls
urun_id	int	<input type="checkbox"/>
urun_ad	nvarchar(100)	<input type="checkbox"/>
fiyat	decimal(6, 2)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Choose Name

?

×

Enter a name for the table:

Urunler

OK

Cancel

Tablolar Arası İlişkiler

DESKTOP-L2N63DT....tDB - dbo.Table_1* X			
	Column Name	Data Type	Allow Nulls
?	alınanurun_id	int	<input type="checkbox"/>
	firma_id	int	<input type="checkbox"/>
	urun_id	int	<input type="checkbox"/>
	tarih	datetime	<input type="checkbox"/>
▶	miktar	float	<input type="checkbox"/>
			<input type="checkbox"/>

Choose Name ? X

Enter a name for the table:

AlinanUrun


Tablolar Arası İlişkiler

DESKTOP-L2N63DT....tDB - dbo.Table_1* ❏ ✕		
Column Name	Data Type	Allow Nulls
PK satilanurun_id	int	<input type="checkbox"/>
firma_id	int	<input type="checkbox"/>
urun_id	int	<input type="checkbox"/>
tarih	datetime	<input type="checkbox"/>
miktar	float	<input type="checkbox"/>
		<input type="checkbox"/>

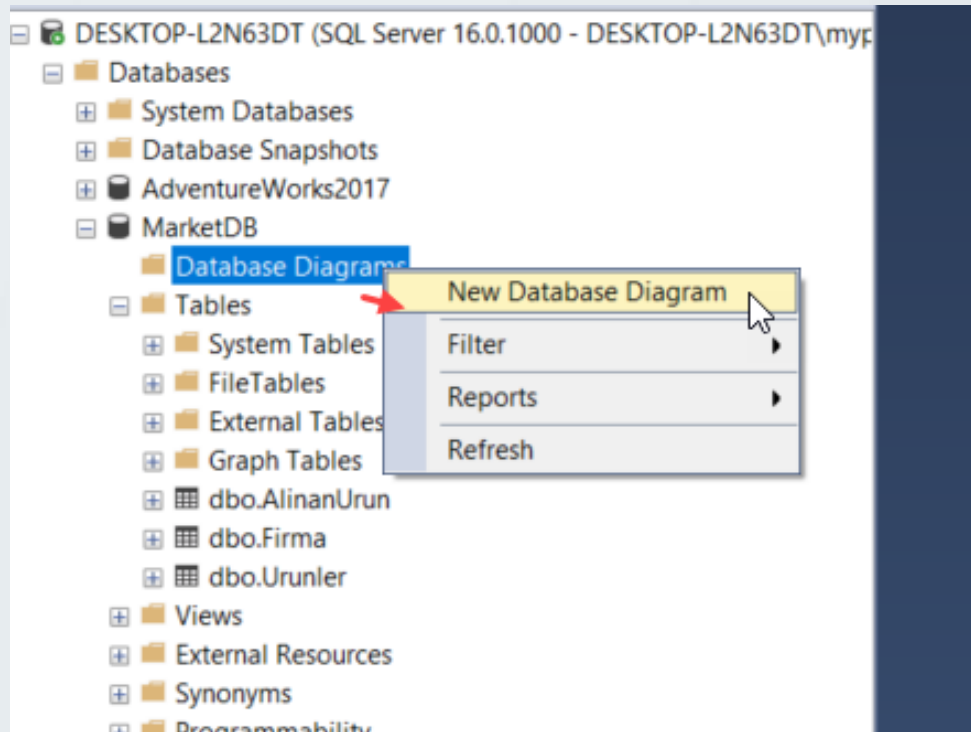
Choose Name ? ✕

Enter a name for the table:

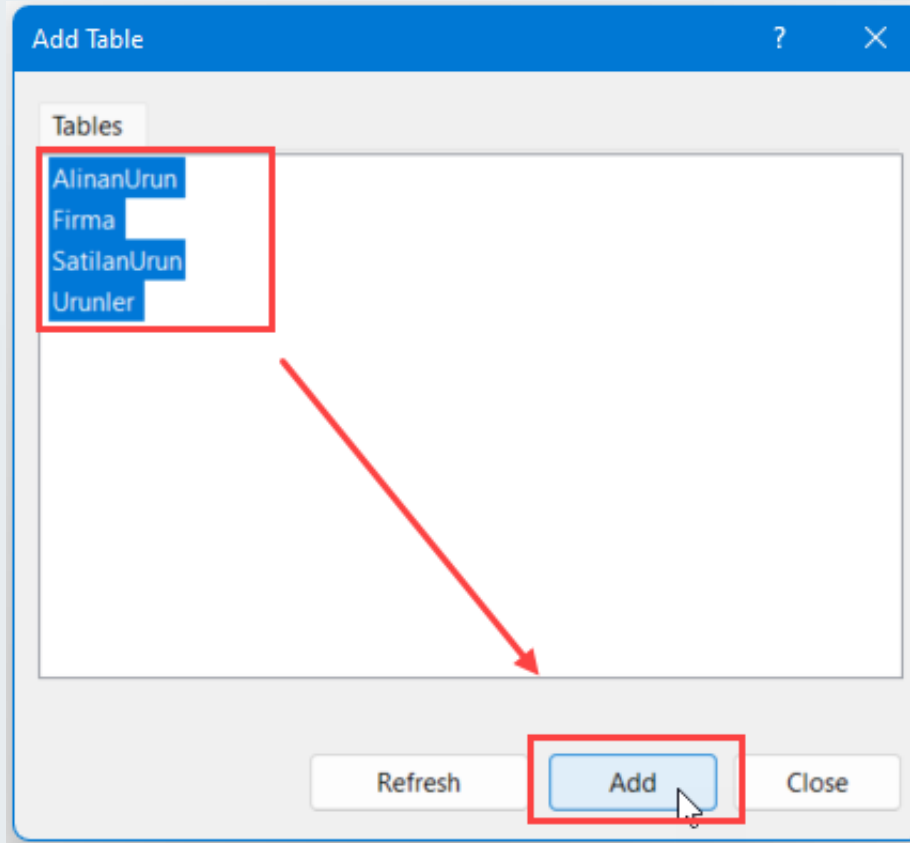
SatilanUrun



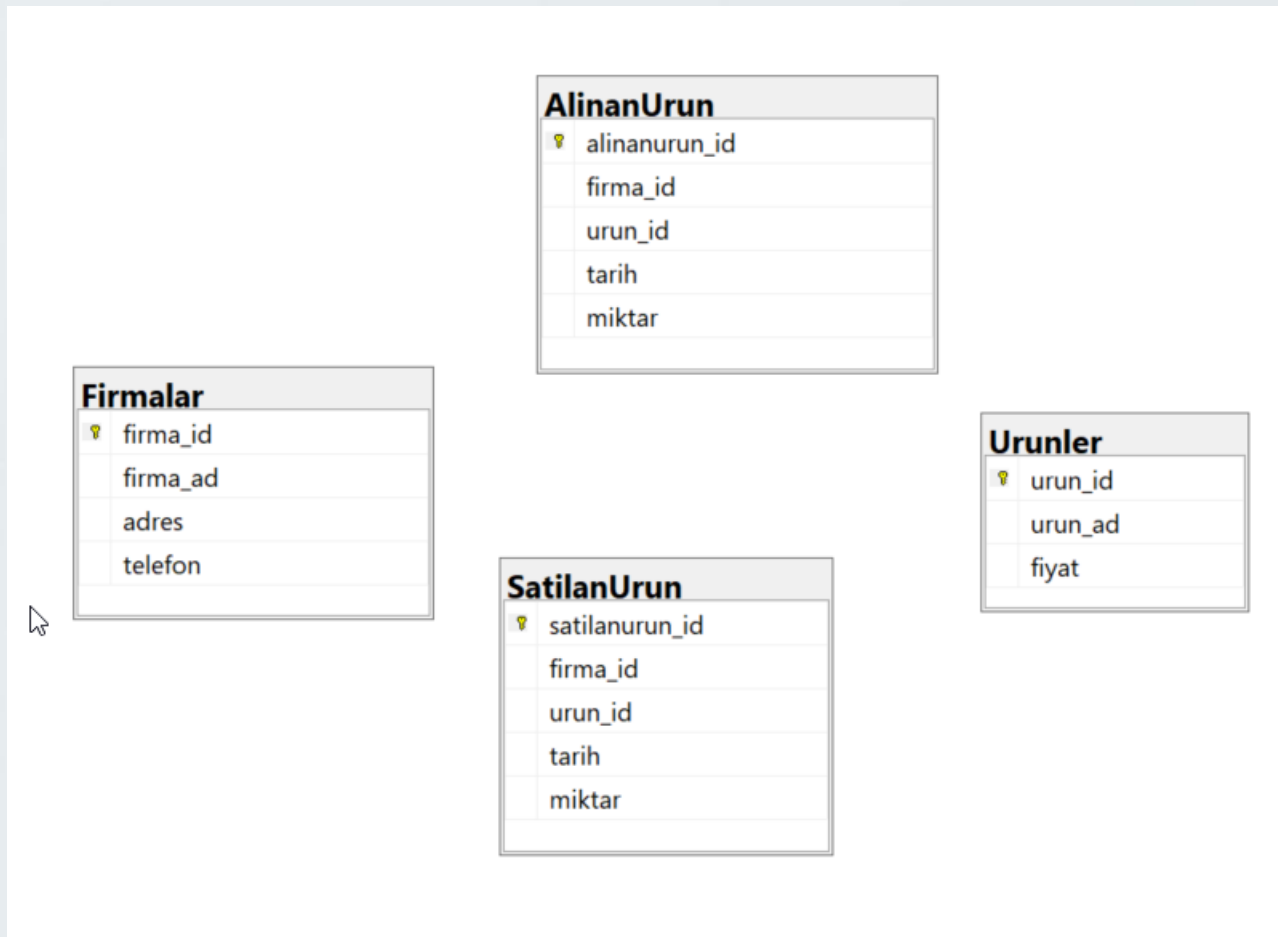
Tablolar Arası İlişkiler



Tablolar Arası İlişkiler



Tablolar Arası İlişkiler



Tablolar Arası İlişkiler

ilişki oluşturma

Firmalar	
	firma_id
	firma_ad
	adres
	telefon

AlinanUrun	
	alinanurun_id
	firma_id
	urun_id
	tarih
	miktar

SatilanUrun

Tablolar Arası İlişkiler

Tables and Columns

Relationship name:
FK_AlınanUrun_Firmalar

Primary key table: Firmalar Foreign key table: AlınanUrun

firma_id firma_id

OK Cancel

Foreign Key Relationship

Selected Relationship:
FK_AlınanUrun_Firmalar*

Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.

(General)
Check Existing Data On Cr Yes
> Tables And Columns Spec
Database Designer
Enforce For Replication Yes
Enforce Foreign Key Cons: Yes
> INSERT And UPDATE Spec
Identity
(Name) FK_AlınanUrun_Firmalar
Description

OK Cancel

Tablolar Arası İlişkiler

