

Tablo oluřturma:

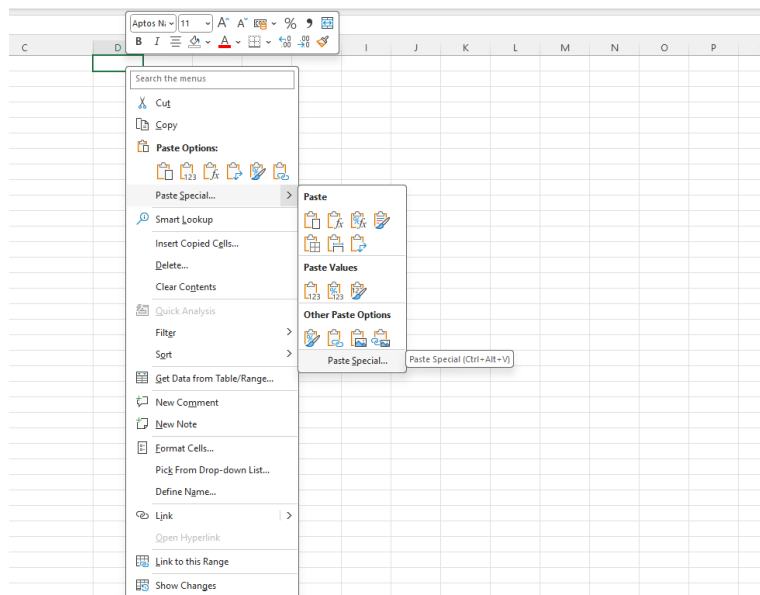
SQL Server ierisinde tablo oluřtururken keyword kelimelerin kolon ismi olarak kullanılmamasına dikkat edilmelidir, eęer kullanılacaksa sonuna _ veya bařka ekler eklenerek o kelimeler deęiřtirilerek kullanılmalıdır.

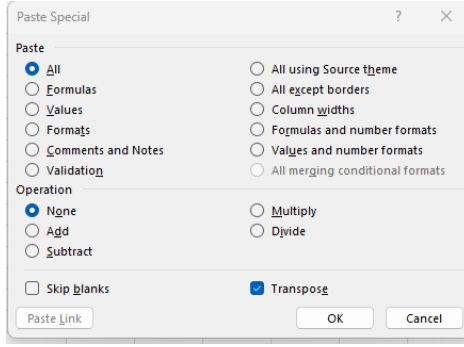
Keywords:

- SELECT
- INSERT
- UPDATE
- DELETE
- USER
- PASSWORD
- GROUP
- ORDER
- TABLE
- WHERE
- JOIN
- FROM

	A	B	C
1	USERS		
2	ID	INT IDENTITY(1,1)	
3	UserName_	varchar(50)	
4	BirthDate	Date	
5	CreatedDate	DateTime	
6	Gender	varchar(1)	
7	NameSurname	varchar(50)	
8	EmailAddress	varchar(50)	
9	Password_	varchar(50)	
10	RecoveryEmail	varchar(50)	
11	Tel	char(10)	
12			
13			

Satırları sutunlara dnřtrerek ierisinde rnek veriler girelim:





Kullanıcıya kargosunu göndermek için adres bilgisine ihtiyaç duyulur, fakat bir kullanıcının birden fazla adresi olabilir, mesela ev adresi, iş adresi, arkadaşının adresine kargosunu isteyebilir. Yani burada 1 to n bir ilişki var, o zaman Addresses olarak ayrı bir tablo olmalı.

ID	CITY	TOWN	DISTRICT	POSTALCODE	ADDRESSTEXT
1	Gümüşhane	Torul	Gülçiçek	29010	Gülçiçek mahallesi, fındık apartmanı, no:8, daire:12

Fakat bir adres içerisinde birden fazla il bulunur ve bu illerin de kendine ait ilçeleri vardır. Mesela Gümüşhane ili için 5 ilçesi olduğundan ötürü 5 kere aşağıdaki gibi yazmak anlamsızdır:

Gümüşhane Torul

Gümüşhane Kelkit

Gümüşhane Köse

Gümüşhane Şiran

Gümüşhane Kürtün

Bu sebepten bir şehirler tablosu da gerekmektedir, çünkü bir şehrin birden fazla ilçesi olabileceği için 1 to n ilişkisi burada da vardır.

ID	CITYNAME
1	Gümüşhane
2	Ardahan

ID	TOWNNAME
1	Torul
1	Kelkit
1	Köse
1	Şiran
1	Kürtün
2	Damal
2	Göle

Master-Detay İlişkisi Kurma:

USERS		ADDRESSES	
ID	INT IDENTITY(1,1)	ID	INT IDENTITY(1,1)
1	kevsr51	1	Gümüşhane
2		2	Ardahan
3		3	
4		4	
5		5	
6		6	
7		7	
8		8	
9		9	
10		10	
11		11	
12		12	
13		13	
14		14	
15		15	
16		16	
17		17	
18		18	
19		19	
20		20	
21		21	
22		22	
23		23	
24		24	

Adres tablosundaki 1 numaralı ID adresinin kime ait olduğunu bilmem için adres tablosuna bir tane kolon daha ekleyelim.

- Yeni kolon USER tablosundan NameSurname bilgisi içerirse, birçok kullanıcı aynı isim ve soyisime sahip olabilir, bu da karışıklığa sebep olacaktır.
 - Yeni kolon USER tablosundan UserName_ bilgisi içerirse, kullanıcı sonradan kullanıcı adını değiştirebilir, bu da kullanıcının adreslerini kaybetmesine neden olur.
- Burada bir şey farketmek gerek, bazı siteler kullanıcı adını bir kere oluşturduktan sonra değiştirmenize izin vermiyor veya kullanıcı mail adresinizi değiştirmenize izin vermiyor, mesela dergipark için arayarak bilgilendirme yapmanız ve maik adresinizi geliştiricilerin değişmesi gerekiyor, belki de bunun sebebi master-detay ilişkisinde primary key olarak adres veya kullanıcı adı kullanılmasından kaynaklıdır.

Sonuç olarak,

Master Table = Users Table (primary key is ID)

Detay Table = Address Table (Foreign key is USERID)

Yani ana tablonun ID sini detay tablosuna ekledik ve ID olarak değil USERID olarak adını değiştirdik:

USERS		ADDRESSES	
ID	INT IDENTITY(1,1)	ID	INT IDENTITY(1,1)
1	kevsr51	1	Gümüşhane
2		2	Ardahan
3		3	
4		4	
5		5	
6		6	
7		7	
8		8	
9		9	
10		10	
11		11	
12		12	
13		13	
14		14	
15		15	
16		16	
17		17	
18		18	
19		19	
20		20	
21		21	
22		22	
23		23	
24		24	

Yukarıdaki adres tablosunda görüldüğü üzere bir kullanıcının birden fazla adresi var.

1. Adres Tablosu için tüm ilişkileri oluşturma:

A	B	C	D	E	F	G	H	I	J	K	L	M
13												
14												
15	ADDRESSES		ID	CITY	TOWN	DISTRICT	POSTALCODE	ADRESSTEXT	USERID			
16	ID	INT IDENTITY(1,1)										
17	CITY	varchar(50)										
18	TOWN	varchar(50)										
19	DISTRICT	varchar(50)										
20	POSTALCODE	char(5)										
21	ADRESSTEXT	varchar(100)										
22	USERID	INT										
23												
24												
25	CITIES		ID	CITYNAME								
26	ID											
27	CITYNAME											
28												
29												
30												
31	TOWNS		ID	TOWNNAME								
32	ID											
33	TOWNNAME											
34												
35												
36												
37												
38												
39												
40												
41												
42												
43												
44	DISTRICT		ID	DISTRICTNAME								
45	ID											
46	DISTRICTNAME											
47												
48												

İlişki oluşturmak için tüm detay tablolarına master tablonun ID sini ekliyorum.

A	B	C	D	E	F	G	H	I	J	K
13										
14										
15	ADDRESSES		ID	CITYID	TOWNID	DISTRICTID	POSTALCODE	ADRESSTEXT	USERID	
16	ID	INT IDENTITY(1,1)								
17	CITYID	varchar(50)								
18	TOWNID	varchar(50)								
19	DISTRICTID	varchar(50)								
20	POSTALCODE	char(5)								
21	ADRESSTEXT	varchar(100)								
22	USERID	INT								
23										
24										
25	CITIES		ID	CITYNAME						
26	ID									
27	CITYNAME									
28										
29										
30										
31	TOWNS		ID	TOWNNAME	CITYID					
32	ID									
33	TOWNNAME									
34	CITYID									
35										
36										
37										
38										
39										
40										
41										
42										
43										
44	DISTRICT		ID	DISTRICTNAME	TOWNID					
45	ID									
46										
47										
48										
49										

İlk görselde her bir tablodaki verilerin ayrı ayrı tutulduğu, **ikinci görselde** ise tablolar arasındaki ilişkilerin kurulduğu ve ID değerlerinin (anahtarların) kullanıldığı bir yapı bulunmaktadır.

Master-Detay İlişkisi:

İkinci görselde, ADDRESSES tablosunda CITYID, TOWNID, ve DISTRICTID alanları kullanılarak CITIES, TOWNS, ve DISTRICT tabloları ile bir ilişki kurulmuş. Bu ilişkide ADDRESSES tablosu "Detay" (child) olarak işlev görürken, CITIES, TOWNS, ve DISTRICT tabloları "Master" (parent) tablolar olarak işlev görmektedir. ADDRESSES tablosunda saklanan her bir satırın, CITIES, TOWNS, ve DISTRICT tablolarındaki ID'lerle eşleştirilmesi sağlanmıştır.

Primary Key - Foreign Key Bağlantısı:

- **Primary Key:** CITIES, TOWNS, ve DISTRICT tablolarındaki ID kolonları, her tablonun birincil anahtarı (Primary Key) olarak tanımlanır. Bu ID'ler her bir kayıt için benzersiz bir değer içerir.

- **Foreign Key:** ADDRESSES tablosundaki CITYID, TOWNID, ve DISTRICTID kolonları ise, bu ID değerlerini referans alır ve "Foreign Key" olarak kullanılır. Bu sayede, ADDRESSES tablosunda tutulan her bir adresin hangi şehirde, hangi kasabada, ve hangi mahallede olduğunu belirtmek için bir ilişki kurulmuş olur.

RDMS Sisteminde Tablolar Arası İlişki:

Bir RDMS (İlişkisel Veritabanı Yönetim Sistemi) içinde, tablolar arasındaki ilişki genellikle **Primary Key (PK)** ve **Foreign Key (FK)** kolonları aracılığıyla kurulur. Bir tablodaki PK, o tablodaki her bir satırı benzersiz kılar. FK ise bir başka tabloda, bir PK kolonuna referans olarak tanımlanır. Bu, iki tablo arasında bir bağlantı kurarak veri tutarlılığını sağlar.

Örnek:

- **CITIES Tablosu:**
 - ID: 1 (PK), CITYNAME: Gümüşhane
 - ID: 2 (PK), CITYNAME: Ardahan
- **TOWNS Tablosu:**
 - ID: 1 (PK), TOWNNAME: Torul, CITYID: 1(FK)
 - ID: 2 (PK), TOWNNAME: Kelkit, CITYID: 1(FK)
 - ID: 3 (PK), TOWNNAME: Göle, CITYID: 2 (FK)
- **ADDRESSES Tablosu:**
 - ID: 1, CITYID: 1, TOWNID: 1, DISTRICTID: 3, ADDRESS: "Gülçiçek Mahallesi"
 - ID: 2, CITYID: 2, TOWNID: 3, DISTRICTID: 7, ADDRESS: "Hasanbey Mahallesi"

Bu örnek, tablolar arasındaki bağlantıların nasıl kurulduğunu gösterir. ADDRESSES tablosunda bir satır eklenirken, o adresin hangi şehre, kasabaya ve mahalleye ait olduğu, CITYID, TOWNID, ve DISTRICTID (foreign keys) kolonları üzerinden belirlenir.

Bu sayede ilişkisel veritabanlarında veri tutarlılığı sağlanmış olur ve veriler birbirine referans verilerek saklanır.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	USERS			ADDRESSES			CITIES			TOWNS			DISTRICT	
2	ID (PK)	INT IDENTITY(1,1)		ID (PK)	INT IDENTITY(1,1)		ID (PK)			ID (PK)			ID (PK)	
3	UserName_	varchar(50)		CITYID (FK)	varchar(50)		CITYNAME			TOWNNAME				
4	BirthDate	Date		TOWNID (FK)	varchar(50)					CITYID				
5	CreatedDate	DateTime		DISTRICTID (FK)	varchar(50)									
6	Gender	varchar(1)		POSTALCODE	char(5)									
7	NameSurname	varchar(50)		ADDRESSTEXT	varchar(100)									
8	EmailAddress	varchar(50)												
9	Password_	varchar(50)												
10	RecoveryEmail	varchar(50)												
11	Tel	char(10)												
12	AddressID (FK)	INT												
13														
14														
15														
16														
17	ITEMS			ORDERS			PAYMENTS							
18	ID (PK)	INT IDENTITY(1,1)		ID (PK)			ID (PK)							
19	ITEMNAME	varchar(50)		DATE_			ORDERID (FK)							
20	BRAND	varchar(50)		TOTALPRICE			ISOK							
21	CATEGORY1	varchar(50)		USERID (FK)			DATE_							
22	CATEGORY2	varchar(50)		ADDRESSID (FK)			APPROVECODE							
23	CATEGORY3	varchar(50)		STATUS_										
24	UNITPRICE	decimal(18,2)		ITEMID (FK)										
25				AMOUNT										
26				UNITPRICE										

SQL SERVER Tablo ve Relation Oluřturma:

SQL Server'da iliřkisel veri tabanı oluřtururken, iki ana dosya t r  kullanılır: **MDF (Master Data File)** ve **Log Dosyası**. Bu dosyalar veri tabanının verilerini ve iřlemelerini y netmek i in kritik  neme sahiptir.

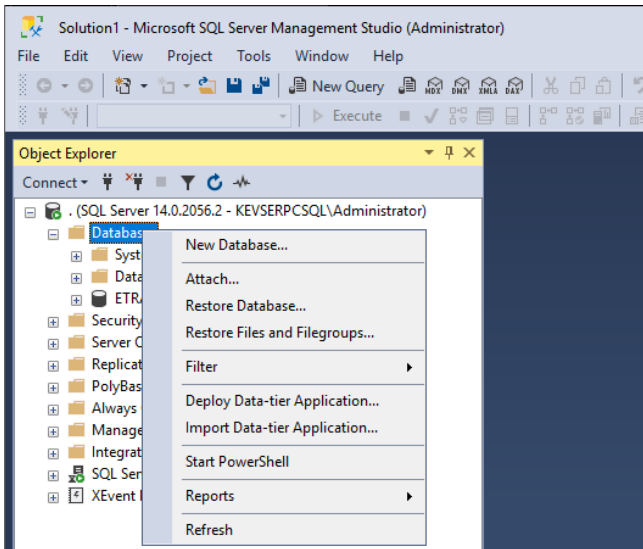
1. MDF (Master Data File)

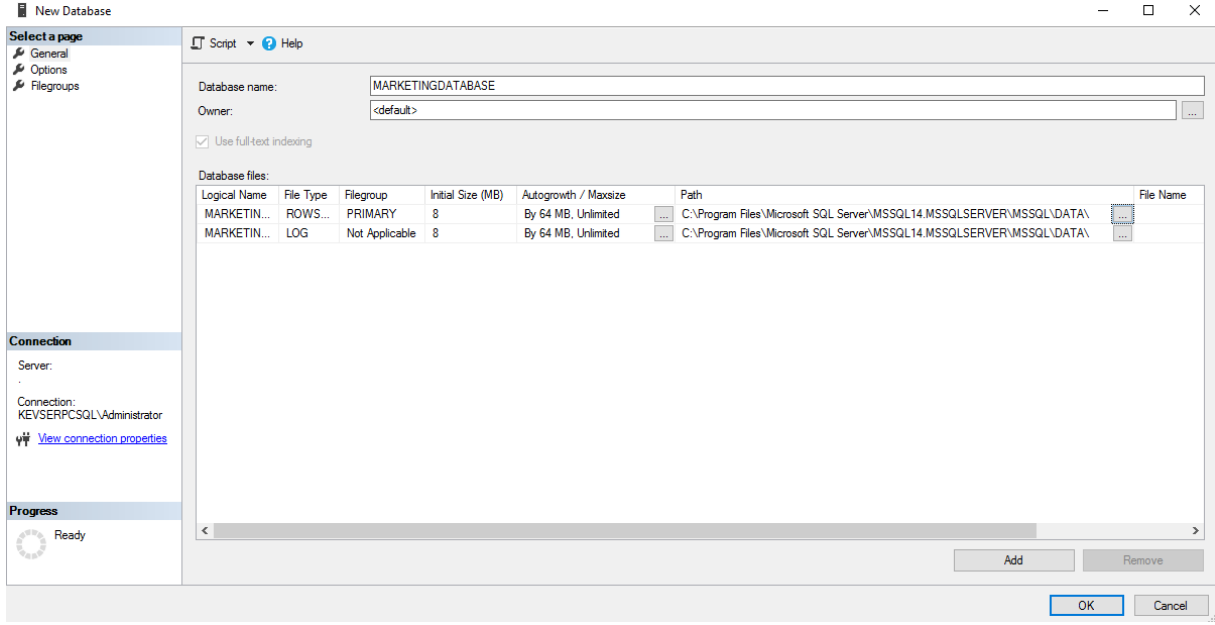
- **Nedir?** MDF dosyası, bir SQL Server veri tabanının ana veri dosyasıdır. Veri tabanındaki t m tablolar, saklı yordamlar (stored procedures), g r n mler (views) ve veri tabanı řeması gibi temel veriler bu dosyada saklanır. Her SQL Server veri tabanında yalnızca bir adet MDF dosyası bulunur.
- **Ne iře Yarar?** MDF dosyası, veri tabanının asıl verilerini i erir. Bu dosya, verilerin fiziksel olarak depolandığı yerdir ve veri tabanındaki t m  nemli verilerin merkezi depolama alanıdır.

2. Log Dosyası

- **Nedir?** Log dosyası, SQL Server'ın veri tabanında yapılan her deėiřikliėi izlediėi ve kaydettiėi dosyadır. Bu dosya, veri tabanındaki iřlemleri (transactions) izler ve geri alma (rollback) veya yeniden yapma (redo) iřlemlerini ger ekleřtirmek i in kullanılır.
- **Ne iře Yarar?** Log dosyası, veri tabanının b t nl ė n  ve tutarlılıė nı saėlamak i in kritik bir rol oynar. Eėer bir iřlem sırasında bir hata oluřursa veya sistem   kmesi yařanırsa, log dosyası bu iřlemi geri alarak veri tabanını tutarlı bir duruma geri getirebilir. Ayrıca, veri tabanının yedekleme (backup) iřlemlerinde de log dosyaları  nemli bir rol oynar,   nk  log dosyaları sayesinde veri tabanı belirli bir zamana geri y klenebilir.

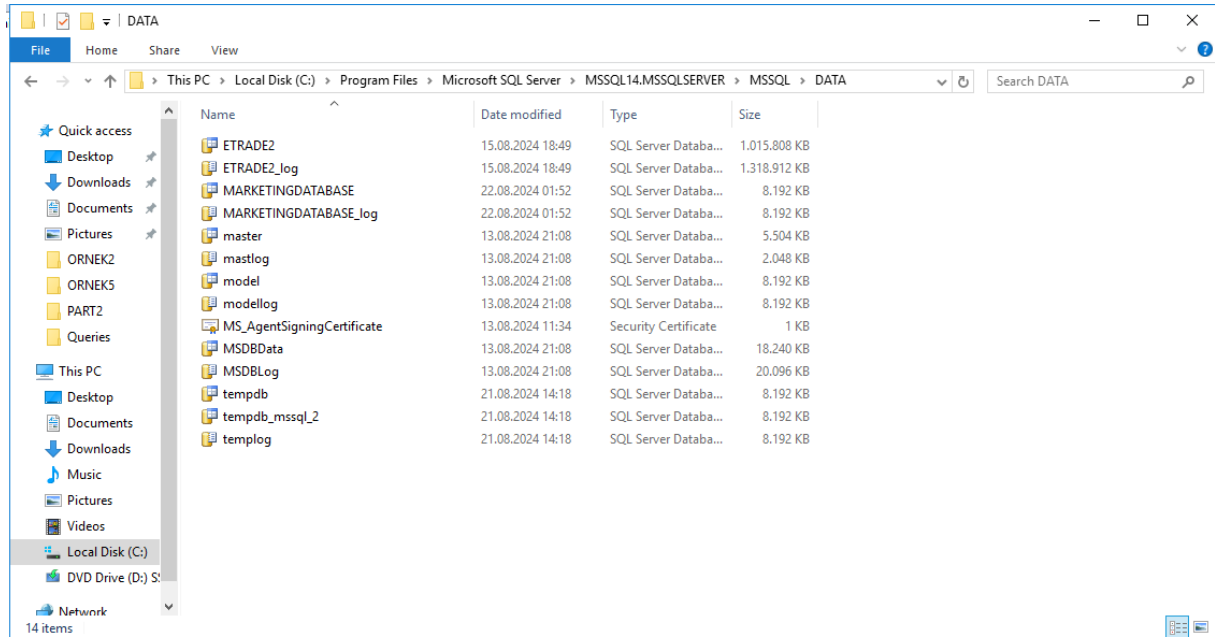
➤ Click new database:





C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\

Eğer dosyalarınıza gidip kontrol etmek isterseniz, oluşturduğunuz database için iki tane MDF ve LOG dosyalarının oluşmuş olduğunu görebilirsiniz:



➤ Daha öncesinde excel tablosunda oluşturduğumuz tabloları SQL Server içerisinde oluşturmaya başlıyoruz:

1. USERS TABLE:

KEVSEPCSQL.MARK...ASE - dbo.Table_1* -> X			
	Column Name	Data Type	Allow Nulls
▶	ID	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Column Properties	
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div></div> <div></div> <div></div> </div> </div>	
▼ (General)	
(Name)	ID
Allow Nulls	Yes
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	
(Is Identity)	No
Identity Increment	
Identity Seed	
Indexable	Yes

Sonrası:

▼ Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

KEVSEPCSQL.MARK...BASE - dbo.USERS -> X			
	Column Name	Data Type	Allow Nulls
▶	ID	int	<input type="checkbox"/>
⚙	Set Primary Key		<input type="checkbox"/>
➕	Insert Column		<input type="checkbox"/>
✖	Delete Column		<input type="checkbox"/>
🔗	Relationships...		<input type="checkbox"/>
🔑	Indexes/Keys...		<input type="checkbox"/>
🔍	Fulltext Index...		<input type="checkbox"/>
📄	XML Indexes...		<input type="checkbox"/>
🔒	Check Constraints...		<input checked="" type="checkbox"/>
📍	Spatial Indexes...		<input type="checkbox"/>
📝	Generate Change Script...		<input type="checkbox"/>
🔧	Properties	Alt+Enter	

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
USERNAME_	varchar(50)	<input checked="" type="checkbox"/>
BIRTHDATE	date	<input checked="" type="checkbox"/>
CREATEDDATE	datetime	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	CREATEDDATE
Allow Nulls	No
Data Type	datetime
Default Value or Binding	

Column Properties	
(General)	
(Name)	CREATEDDATE
Allow Nulls	No
Data Type	datetime
Default Value or Binding	GETDATE()

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
USERNAME_	varchar(50)	<input type="checkbox"/>
BIRTHDATE	date	<input type="checkbox"/>
CREATEDDATE	datetime	<input type="checkbox"/>
GENDER	varchar(1)	<input type="checkbox"/>
NAMESURNAME	varchar(50)	<input type="checkbox"/>
EMAIL	varchar(50)	<input type="checkbox"/>
PASSWORD_	varchar(50)	<input type="checkbox"/>
RECOVERYEMAIL	varchar(50)	<input checked="" type="checkbox"/>
PHONENUMBER	char(10)	<input type="checkbox"/>

2. ADDRESS Tabloları:

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
USERID	int	<input type="checkbox"/>
CITYID	smallint	<input type="checkbox"/>
TOWNID	int	<input type="checkbox"/>
DISTRICTID	int	<input type="checkbox"/>
POSTALCODE	char(5)	<input type="checkbox"/>
ADDRESSTEXT	varchar(300)	<input type="checkbox"/>

KEVSERPCSQL.MAR...BASE - dbo.CITIES			
	Column Name	Data Type	Allow Nulls
PK	ID	smallint	<input type="checkbox"/>
	CITYNAME	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

KEVSERPCSQL.MAR...ASE - dbo.TOWNS			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	CITYID	smallint	<input type="checkbox"/>
	TOWNNAME	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

KEVSERPCSQL.MAR... - dbo.DISTRICTS			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	TOWNID	int	<input type="checkbox"/>
	DISTRICTNAME	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

3. Orders and Payment Tabloları

KEVSERPCSQL.MAR...ASE - dbo.ORDERS			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	USERID_	int	<input type="checkbox"/>
	ADDRESSID	int	<input type="checkbox"/>
	ITEMID	int	<input type="checkbox"/>
	DATE_	datetime	<input type="checkbox"/>
	AMOUNT	tinyint	<input type="checkbox"/>
	UNITPRICE	decimal(18, 2)	<input type="checkbox"/>
	TOTALPRICE	decimal(18, 2)	<input type="checkbox"/>
	STATUS_	tinyint	<input type="checkbox"/>
			<input type="checkbox"/>

KEVSERPCSQL.MAR...E - dbo.PAYMENTS			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	ORDERID	int	<input type="checkbox"/>
	DATE_	datetime	<input type="checkbox"/>
	TOTALPRICE	decimal(18, 2)	<input type="checkbox"/>
	ISOK	bit	<input type="checkbox"/>
	APPROVECODE	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Relation Atama (FK ve PK arasında ilişki kurma) :

İlişkisel veritabanı kurmak iki türlü olabilir:

- Yazılımcıların FK-PK prensibine dayanarak kendi kontrollerinde olabilir
- Veri tabanında relation tanımlayarak olabilir.

Bazı şirketler veri tabanında ilişki kurmanın yazılımcıyı kısıtlayan bir durum olduğunu düşünerek relation ataması yapmazlar ve geliştiriciler bu anahtarlara dikkat ederek veri girişi yapar.

Kullanım tamamen geliştiriciye kalmıştır.

1. Yazılımcıların FK-PK prensibine dayalı veri girmeleri:

ID	CITYNAME
1	Trabzon
2	Gümüşhane
3	Ardahan
4	Ağrı
5	Rize
6	Aydın
7	Muğla
8	Antalya
9	Van
10	Bitlis
11	Kars
12	Ankara
13	Kocaeli
14	Giresun
15	Artvin
16	Erzurum
17	Erzincan
NULL	NULL

ID	CITYID	TOWNNAME
1	3	Göle
2	3	Hanak
3	3	Damal
5	6	Nazilli
6	6	Söke
7	6	Köşk
8	6	Didim
9	6	Germencik
10	1	Ortahisar
11	1	Maçka
12	1	Şalpazarı
13	16	Oltu
14	16	İspir
15	16	Tortum
16	9	Çaldıran
17	9	Özalp
NULL	NULL	NULL

```
SQLQuery1.sql - (lo...Administrator (59)) * X KEYSERPCSQL.MAR...ASE - dbo.TOWNS KEYSERPCSQL.M...
SELECT *
FROM CITIES
DELETE CITIES WHERE ID=6
```

100 %

Messages

(1 row affected)

SELECT *
FROM CITIES

100 %

	ID	CITYNAME
1	1	Trabzon
2	2	Gümüşhane
3	3	Ardahan
4	4	Ağrı
5	5	Rize
6	7	Muğla
7	8	Antalya
8	9	Van
9	10	Bitlis
10	11	Kars
11	12	Ankara
12	13	Kocaeli
13	14	Giresun
14	15	Artvin
15	16	Erzurum

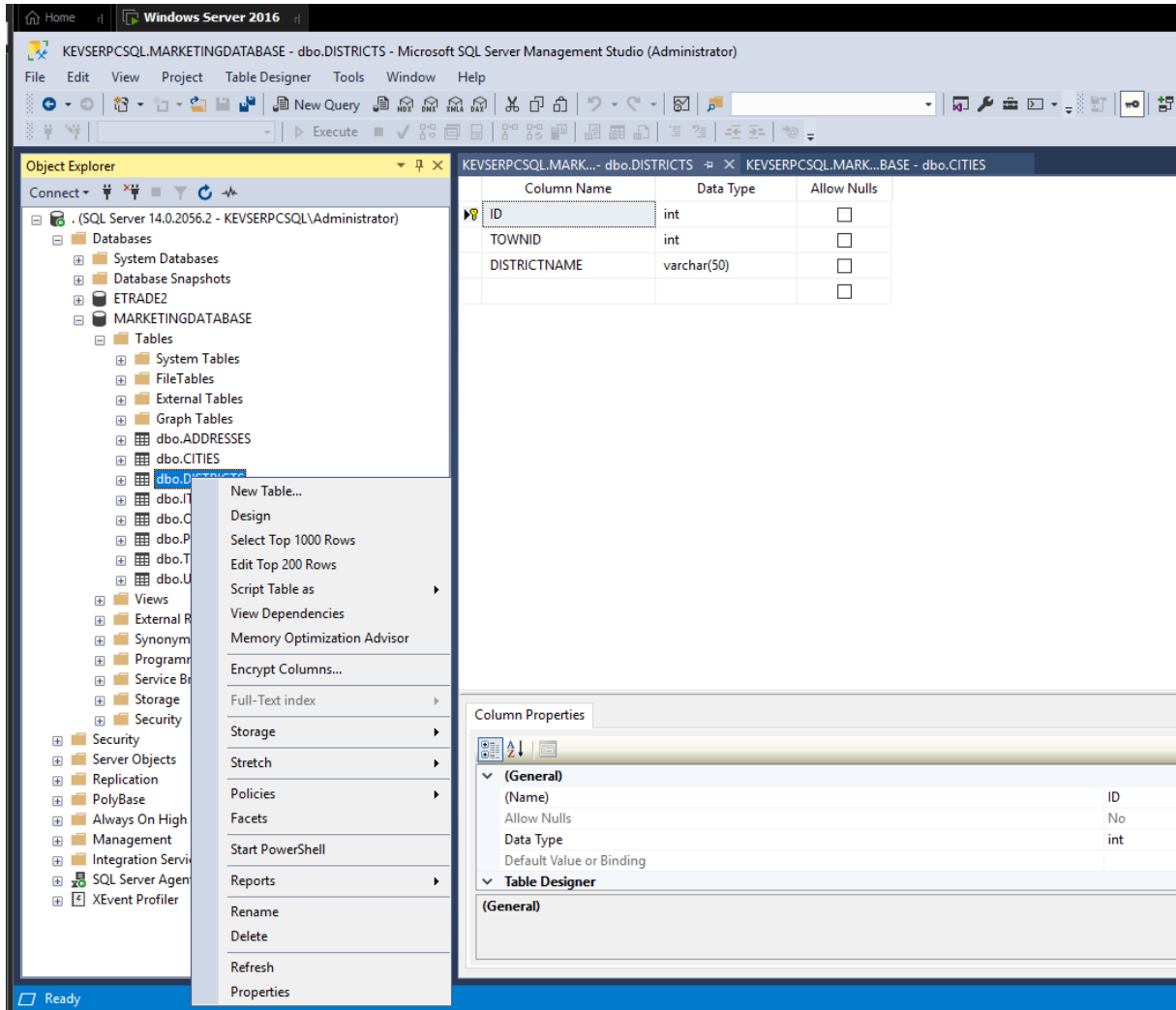
SELECT *
FROM TOWNS

100 %

	ID	CITYID	TOWNNAME
2	2	3	Hanak
3	3	3	Damal
4	5	6	Nazilli
5	6	6	Söke
6	7	6	Köşk
7	8	6	Didim
8	9	6	Gemencik
9	10	1	Ortahisar
10	11	1	Mağca
11	12	1	Şalpazarı
12	13	16	Oltu
13	14	16	İspir
14	15	16	Tortum
15	16	9	Çaldıran
16	17	9	

Query executed successfully.

2. Veri tabanında relation tanımlayarak olabilir:



Microsoft SQL Server Management Studio (Administrator) - KEYSERPCSQL.MARKETINGDATABASE - dbo.DISTRICTS

Object Explorer: (SQL Server 14.0.2056.2 - KEYSERPCSQL\Administrator)

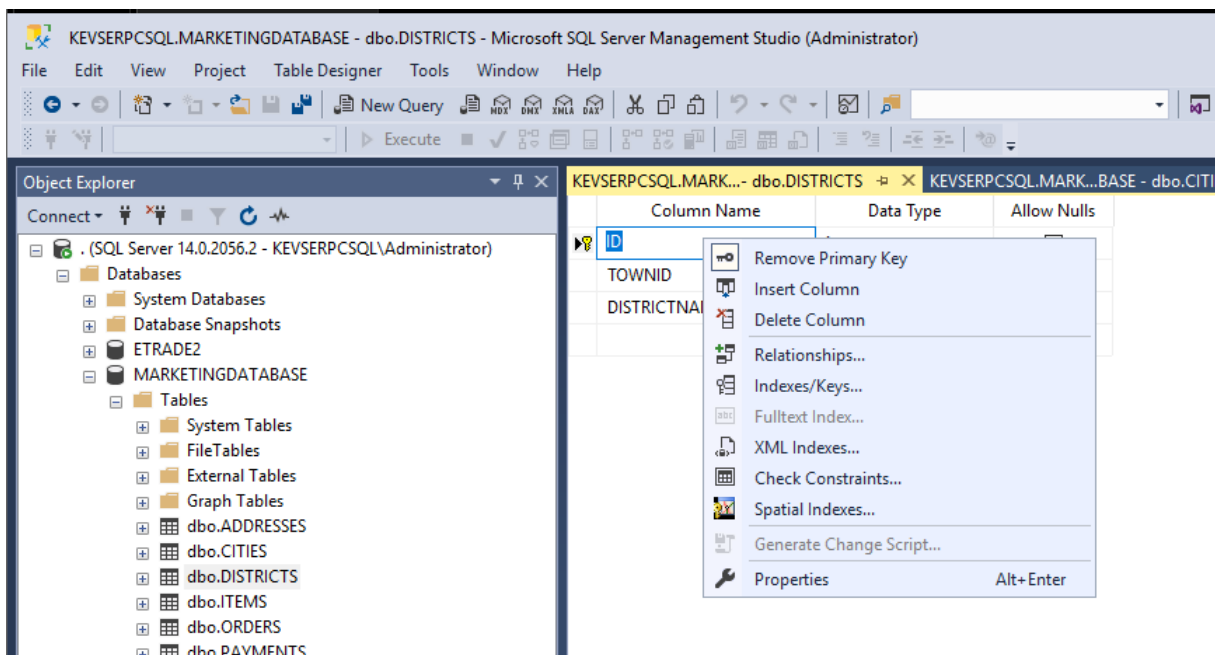
- Databases
 - System Databases
 - Database Snapshots
 - ETRADE2
 - MARKETINGDATABASE
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.ADDRESSES
 - dbo.CITIES
 - dbo.DISTRICTS
 - dbo.ITEMS
 - dbo.ORDERS
 - dbo.PAYMENTS
 - Views
 - External R
 - Synonym
 - Program
 - Service Br
 - Storage
 - Security

Table Designer: KEYSERPCSQL.MARK... - dbo.DISTRICTS

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
TOWNID	int	<input type="checkbox"/>
DISTRICTNAME	varchar(50)	<input type="checkbox"/>

Column Properties: (General)

Property	Value
(Name)	ID
Allow Nulls	No
Data Type	int
Default Value or Binding	



Microsoft SQL Server Management Studio (Administrator) - KEYSERPCSQL.MARKETINGDATABASE - dbo.DISTRICTS

Object Explorer: (SQL Server 14.0.2056.2 - KEYSERPCSQL\Administrator)

- Databases
 - System Databases
 - Database Snapshots
 - ETRADE2
 - MARKETINGDATABASE
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.ADDRESSES
 - dbo.CITIES
 - dbo.DISTRICTS
 - dbo.ITEMS
 - dbo.ORDERS
 - dbo.PAYMENTS
 - Views
 - External R
 - Synonym
 - Program
 - Service Br
 - Storage
 - Security

Table Designer: KEYSERPCSQL.MARK... - dbo.DISTRICTS

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
TOWNID	int	<input type="checkbox"/>
DISTRICTNAME	varchar(50)	<input type="checkbox"/>

Table Designer Context Menu:

- Remove Primary Key
- Insert Column
- Delete Column
- Relationships...
- Indexes/Keys...
- Fulltext Index...
- XML Indexes...
- Check Constraints...
- Spatial Indexes...
- Generate Change Script...
- Properties (Alt+Enter)

Foreign Key Relationships

Selected Relationship:

FK_DISTRICTS_DISTRICTS*

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 Create: Yes
 Tables And Columns Specification: ...
 Foreign Key Base Table: DISTRICTS
 Foreign Key Columns: ID
 Primary/Unique Key Base: DISTRICTS
 Primary/Unique Key Column: ID
 Identity
 (Name): FK_DISTRICTS_DISTRICTS
 Description:
 Table Designer

Add Delete Close

Tables and Columns

Relationship name:

FK_DISTRICTS_TOWNS

Primary key table: TOWNS Foreign key table: DISTRICTS

ID TOWNID

OK Cancel

Foreign Key Relationships

Selected Relationship:

FK_DISTRICTS_TOWNS*

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

Primary/Unique Key Base: TOWNS
 Primary/Unique Key Column: ID
 Identity
 (Name): FK_DISTRICTS_TOWNS
 Description:
 Table Designer
 Enforce For Replication: Yes
 Enforce Foreign Key Constraint: Yes
 INSERT And UPDATE Specific:
 Delete Rule: No Action
 Update Rule: No Action

Add Delete Close

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
CITYID	smallint	<input type="checkbox"/>
TOWNNAME		

Relationship name: FK_TOWNS_CITIES

Primary key table: CITIES Foreign key table: TOWNS

ID CITYID

OK Cancel

NOT: İlişki oluşturmadan önce tablolarda dikkat edilmesi gereken şeyler vardır:

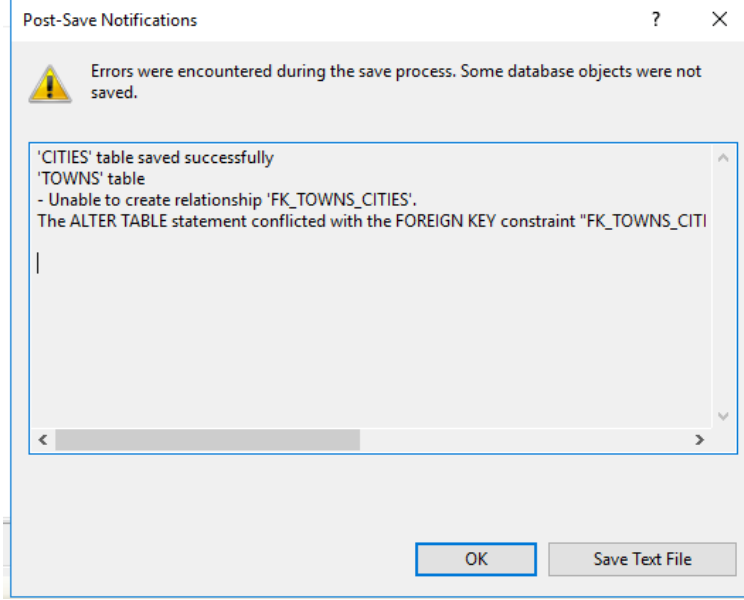
1. Tablolar arası ilişki için her tablonun bir primary key olmak zorunda.
2. Tablo sütunlarının aynı veri tipine ve uzunluğa sahip olması gerekmektedir.

Column Name	Data Type	Allow Nulls
ID	smallint	<input type="checkbox"/>
CITYNAME	varchar(50)	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
CITYID	smallint	<input type="checkbox"/>
TOWNNAME	varchar(50)	<input type="checkbox"/>

3. Tablolarda veri tabansal ilişki oluşturmadan önce geliştiricinin ilişkisel veri girdiği sütun verileri silinirse, diğer tablo etkileneneğinden ilişkili ID nin silinip silinmediği kontrol edilmelidir.

HATA ! cities ve towns tablolarım yüklediğim gibi ve relation da yüklediğim gibi. relation oluşturduktan sonra tabloyu kaydederken böyle bir hata alıyorum:



'CITIES' table saved successfully

'TOWNS' table

- Unable to create relationship 'FK_TOWNS_CITIES'.

The ALTER TABLE statement conflicted with the FOREIGN KEY constraint "FK_TOWNS_CITIES".
The conflict occurred in database "MARKETINGDATABASE", table "dbo.CITIES", column 'ID'.

ÇÖZÜM !

Bu hata mesajı genellikle mevcut veri değerlerinin uyumsuzluğu nedeniyle ortaya çıkar. İşte bu duruma neden olabilecek bazı yaygın sebepler ve çözüm önerileri:

1. Veri Uyuşmazlığı:

- **Açıklama:** Eğer TOWNS tablosundaki CITYID sütununda, CITIES tablosundaki ID sütununda bulunmayan bir değer varsa, bu ilişkiyi oluştururken hata alırsın. Örneğin, CITYID 10 değerine sahip bir satır varsa ve CITIES tablosunda ID 10 değerine sahip bir satır yoksa, bu hata mesajını alırsın.
- **Çözüm:** TOWNS tablosundaki CITYID değerlerinin CITIES tablosundaki ID değerleriyle uyumlu olduğundan emin olmalısın. Uyuşmayan CITYID değerlerini sorgulamak için aşağıdaki SQL sorgusunu kullanabilirsin:

```
SELECT CITYID  
FROM TOWNS  
WHERE CITYID NOT IN (SELECT ID FROM CITIES);
```


2. Boş Değerler (Null):

- **Açıklama:** Eğer TOWNS tablosundaki CITYID sütununda NULL değerleri varsa ve yabancı anahtar (Foreign Key) kısıtlaması bu değerlere izin vermiyorsa, bu hata mesajını alabilirsin.
- **Çözüm:** TOWNS tablosundaki CITYID sütunundaki NULL değerleri kontrol etmelisin. NULL değerler varsa ve bunların izin verilip verilmediğini kontrol etmelisin. Aşağıdaki sorgu ile NULL değerleri kontrol edebilirsin:

```
SELECT *  
FROM TOWNS  
WHERE CITYID IS NULL;
```

3. Veri Türü Uyumsuzluğu:

- **Açıklama:** CITIES tablosundaki ID sütunu ile TOWNS tablosundaki CITYID sütununun veri türleri uyumlu olmalı. Her iki sütun da smallint olarak tanımlanmış ve bu doğru bir yapılandırma gibi görünüyor.
- **Çözüm:** Veri türü uyumsuzluğu ihtimali düşük, çünkü her iki sütun da smallint olarak tanımlanmış.

4. Tablo Boş Olması:

- **Açıklama:** Eğer CITIES tablosu boşsa, yani henüz hiçbir satır eklenmediyse ve TOWNS tablosunda CITYID değerleri varsa, ilişki oluşturulurken hata alırsın.
- **Çözüm:** CITIES tablosunun boş olmadığını ve en azından TOWNS tablosundaki CITYID değerleri ile eşleşen ID değerleri olduğunu kontrol etmelisin. Aşağıdaki sorgu ile CITIES tablosunun boş olup olmadığını kontrol edebilirsin:

```
SELECT * FROM CITIES;
```

5. Yabancı Anahtar Kısıtlamasını Geçici Olarak Devre Dışı Bırakmak:

- Eğer yukarıdaki adımlardan birinde hatalı veya eksik bir veri tespit edersen, bu verileri düzeltmeden önce yabancı anahtar kısıtlamasını geçici olarak devre dışı bırakabilirsin. Ancak, bu yalnızca geçici bir çözüm olup, veri tutarlılığı sağlanmadan kalıcı çözüm olarak kullanılmamalıdır.

ÇÖZÜMÜN UYGULANMASI !

```
SELECT CITYID  
FROM TOWNS  
WHERE CITYID NOT IN (SELECT ID FROM CITIES);
```

100 %

Results Messages

	CITYID
1	6
2	6
3	6
4	6
5	6

Kontrol ediyorum: CITIES tablosundaki 6 numaralı ID ye sahip şehir silinmiş ve town tablosundaki AYDIN ilçeleri sahipsiz kalmış:

ID	CITYNAME
1	Trabzon
2	Gümüşhane
3	Ardahan
4	Ağrı
5	Rize
7	Muğla
8	Antalya
9	Van
10	Bitlis
11	Kars
12	Ankara
13	Kocaeli
14	Giresun
15	Artvin
16	Erzurum
17	Erzincan
*	NULL

ID	CITYID	TOWNNAME
1	3	Göle
2	3	Hanak
3	3	Damal
5	6	Nazilli
6	6	Söke
7	6	Köşk
8	6	Didim
9	6	Germencik
10	1	Ortahisar
11	1	Maçka
12	1	Şalpazarı
13	16	Oltu
14	16	İspir
15	16	Tortum
16	9	Çaldıran
17	9	Özalp
*	NULL	NULL

Towns tablosundaki CITYID=6 olan tüm verileri sildik çözüm olarak:

ID	CITYID	TOWNNAME
1	3	Göle
2	3	Hanak
3	3	Damal
10	1	Ortahisar
11	1	Maçka
12	1	Şalpazarı
13	16	Oltu
14	16	İspir
15	16	Tortum
16	9	Çaldıran
17	9	Özalp
*	NULL	NULL

➤ ADDRESS tablosu için relation oluşturuluyor:

KEVSERPCSQL.MARK...- dbo.ADDRESSES			
KEVSERPCSQL.MARK...BASE - db			
	Column Name	Data Type	Allow Nulls
PK	ID	int	<input type="checkbox"/>
	USERID	int	<input type="checkbox"/>
	CITYID	smallint	<input type="checkbox"/>
	TOWNID	int	<input type="checkbox"/>
	DISTRICTID	int	<input type="checkbox"/>
	POSTALCODE	char(5)	<input type="checkbox"/>
	ADDRESSTEXT	varchar(300)	<input type="checkbox"/>
			<input type="checkbox"/>

Tables and Columns

Relationship name: FK_ADDRESSES_CITIES

Primary key table: CITIES Foreign key table: ADDRESSES

ID CITYID

OK Cancel

Tables and Columns

Relationship name: FK_ADDRESSES_TOWNS

Primary key table: TOWNS Foreign key table: ADDRESSES

ID TOWNID

OK Cancel

Tables and Columns

Relationship name:
FK_ADDRESSES_DISTRICTS

Primary key table:
DISTRICTS

Foreign key table:
ADDRESSES

ID DISTRICTID

OK Cancel

Tables and Columns

Relationship name:
FK_ADDRESSES_USERS

Primary key table:
USERS

Foreign key table:
ADDRESSES

ID USERID

OK Cancel

KEVSEPCSQL.MARK... - dbo.ADDRESSES KEVSEPCSQL.MARK... - BASE - dbo.CITIES KEVSEPCSQL.MARK... - dbo.DISTRICTS

Column Name	Data Type
ID	int
USERID	int
CITYID	smallint
TOWNID	int
DISTRICTID	int
POSTALCODE	char(5)
ADDRESSTEXT	varchar(300)

Foreign Key Relationships

Selected Relationship:

- FK_ADDRESSES_CITIES*
- FK_ADDRESSES_DISTRICTS*
- FK_ADDRESSES_TOWNS*
- FK_ADDRESSES_USERS*

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 Create: Yes

Tables And Columns Specific:

Identity

(Name) FK_ADDRESSES_USERS

Description

Table Designer

Enforce For Replication: Yes

Enforce Foreign Key Constraint: Yes

INSERT And UPDATE Specific:

Add Delete Close

DDL ile Tablo Oluşturma:

Scriptile ile Tablo Oluşturma:

References:

<https://tayipturk.medium.com/sql-de-dml-dcl-ddl-dql-nedir-83f19e02da6d>

[Uygulamalarla SQL Öğreniyorum \(btkakademi.gov.tr\)](https://www.btkakademi.gov.tr/uygulamalarla-sql-ogreniyorum)