بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

مقدمه ای بر پایگاه داده ی رابطه ای (relational database)

مدرس : مهندس افشین رفوآ

دوره آموزشی Sql Server

مقدمه ای بر پایگاه داده ی رابطه ای (relational database)

قید کلید اصلی (**Primary Key Constraint**)

بانک اطلاعاتی رابطه ای: بانک اطلاعاتی یا سیستم مدیریت بانک اطلاعاتی که داده ها را در جدول ذخیره می کند – سطرها و ستون هایی از داده ها – با استفاده از داده های ستون های یک جدول، داده های دیگری را در یک جدول دیگر جستجو و پیدا می کند. سطرهای یک جدول در بانک اطلاعاتی رابطه ای نمایانگر رکوردها و ستون های آن نیز نمایانگر فیلدها هستند. پایگاه داده ی رابطه ای در حین جستجو، اطلاعات فیلدی از یک جدول را با اطلاعات فیلد متناظرش در یک جدول دیگر تطبیق داده و با ترکیب داده های درخواستی از هر دو جدول، یک جدول سوم را تولید می کند. به عبارت دیگر یک بانک اطلاعاتی رابطه ای از مقادیر متناظر دو جدول برای مرتبط کردن اطلاعات یک جدول یا یک جدول دیگر استفاده می کند.

پایگاه داده رابطه‌ای (Relational databases) به آن دسته از پایگاه‌های داده اطلاق می‌شود که بر اساس مدل رابطه‌ای طراحی و ایجاد شده باشند. پایگاه داده ی رابطه ای در واقع یک سیستم است که رابطه میان اشیا چند پایگاه داده را مشخص می کند. به عنوان مثال می توان در پایگاه داده ی یک بانک، با استفاده از یک شی برای مشتریان حساب باز کرده و با استفاده از شی دیگر تراکنش هایی که صاحبان حساب بانکی به آن نیاز دارند را پردازش کرد. دلیلش این است که یک مشتری ممکن است نیاز داشته باشد چندین تراکنش مختلف به طور مکرر انجام دهد. بجای اینکه هر بار مشتری می خواهد یک تراکنش جدید اجرا کند، حساب جدید ایجاد کنیم، می توان یک حساب به عنوان مرجع ایجاد کرد و سپس هر بار که مشتری می خواهد مبلغی را واریز کرده یا از آن بیرون بکشد، همین حساب را فراخوانی کنیم.

برای اعمال قوانین پایگاه داده ی رابطه ای، بایستی نوعی رابطه بین اشیا پایگاه داده برقرار کرد. تراکنش هایی که بین اشیا مختلف پایگاه داده انجام می شود، باید اطمینان حاصل کند اطلاعات یک شی برای شی دیگر قابل دسترسی باشد. اشیایی که آن اطلاعات را ذخیره دارند، همان جداول هستند.

برای مدیریت جریان اطلاعات از یک جدول (A) به جدول دیگر (B)، جدولی که حاوی اطلاعات است (A)، باید آن را در اختیار دیگر جداول قرار دهد (برای مثال B). البته در این میان مسائلی هست که باید به آن توجه داشت:

1. باید این امکان وجود داشته باشد که هر رکورد را به طور منحصر بفرد و بدون سردرگمی شناسایی کرد. برای مثال، اگر فهرستی از اتومبیل ها در یک جدول ایجاد کنید، باید مطمئن شوید یک شماره ی یکتا و غیرتکراری برای هر اتومبیل وجود دارد. این کار باعث می شود رکورد تکراری در جدول وجود نداشته باشد.

2. جدل حاوی اطلاعات لازم است اطلاعات خود را در اختیار جداول دیگر (B) قرار دهد.

3. دو جدول نباید دربردارنده ی اطلاعات یکسان باشند. زمانی که اطلاعات منحصر بفرد را در هر جدول وارد کردید، یک جدول می تواند اطلاعات خود را در اختیار دیگر جداول موجود در پایگاه داده که به آن احتیاج دارند، قرار دهد تا بدین وسیله اطلاعات یک جدول در جدول دیگر عینا تکرار نشود.

تمامی این مشکلات از طریق مشخص کردن یک ستون یا گروهی از ستون ها تحت عنوان کلید یا "key" جدول، قابل حل می باشد. این ستون یا ستون ها در اصطلاح primary key (کلید اصلی) خوانده می شوند.

cدر یک پایگاه داده ی رابطه ای، هر جدول باید حداقل یک کلید اصلی داشته باشد. مثال: می توان یک کلید اصلی در جدول Customers پایگاه داده ی بانک بر روی فیلد Bank Account اعمال کرد زیرا هر مشتری باید یک شماره حساب منحصر بفرد و غیر تکراری داشته باشد.

پس از تصمیم به اعمال محدودیت کلید اصلی برای جدول، باید نوع داده و اطلاعاتی که ستون کید اصلی جدول می پذیرند را مشخص نمایید. مثالی در این زمنیه ستونی حاوی مقدار شماره ی قفسه ی کتابخانه در جدول است که نوع داده ای آن را char، nchar ، varchar یا nvarchar تنظیم کرده و سپس ستون ذکر شده را به عنوان کلید اصلی آن جدول انتخاب کنیم.(پس از اینکه تصمیم گرفتید جدول باید یک فیلد کلید اصلی داشته باشد، بایستی نوع داده ای آن فیلد را تعریف کنید (مشخص نمایید ستون کلید اصلی چه نوع داده هایی را در خود ذخیره کند). برای مثال یک جدول ایجاد می کنید به نام library و بعد ستونی که حاوی مقدار شماره ی قفسه کتابخانه است را به عنوان کلید اصلی انتخاب کنید اما پیش از آن نوع داده ای ستون کلید اصلی را char یا مشتقات آن تنظیم نمایید).

نوع داده ای ستون کلید اصلی غالبا int می باشد.

یکی از اصولی که باید در تعریف کلید اصلی درنظر داشته باشید، این است که هر جدول باید فقط یک کلید اصلی داشته باشد. اگر یک ستون را به عنوان کلید اصلی انتخاب کنید، آن ستون به هیچ وجه نمی تواند مقدار null داشته باشد.

آموزش کاربردی

1. محیط Microsoft SQL Server را راه اندازی کرده و دکمه ی Connect را در پنجره ی محاوره ی اتصال به سرور کلیک کنید.

2. بر روی اسم server راست کلیک کرده و سپس گزینه ی New Query را انتخاب نمایید.

3. برای ایجاد یک پایگاه داده ی جدید، کد زیر را وارد نمایید:

```
CREATE DATABASE RealEstate1;
GO
USE RealEstate1;
GO
CREATE SCHEMA Listing;
GO
```

4. برای ایجاد یک پایگاه داده ی جدید، در منوی اصلی: Query -> Execute.

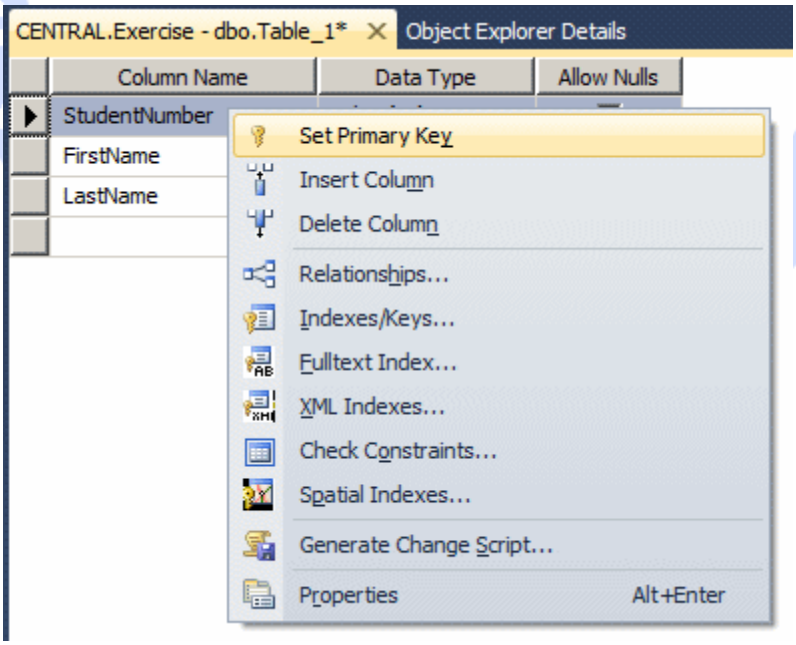
ایجاد کلید اصلی به صورت ویزاردی

به منظور ایجاد یک محدودیت کلید اصلی در محیط مدیریت بانک اطلاعاتی Microsoft SQL Server، یک ستون ایجاد کرده و نوع داده ای آن را تعریف نمایید:

1. سپس، در نوار ابزار بر روی دکمه ی  کلیک نمایید.

2. یا می توانید بر روی ستون مورد نظر راست کلیک کرده و گزینه ی Set Primary Key را انتخاب نمایید.

مثال:



ایجاد کلید اصلی از طریق کدنویسی

1. به منظور ایجاد یک ستون با دستورات SQL، کافی است در سمت راست تعریف ستون، کلیدواژه ی PRIMARY KEY را درج نمایید. مثال:

```
CREATE TABLE Persons
(
   PersonID int PRIMARY KEY NOT NULL,
   FirstName nvarchar(20),
   LastName nvarchar(20) NOT NULL
);
```

ایجاد یک محدودیت کلید اصلی (Primary Key Constraint)

در SQL می توان یک اسم مشخص به کلید اصلی اختصاص داد. برای این منظور، ابتدا ستون را ایجاد می کنیم. سپس، پیش از کلیدواژه ی PRIMARY KEY و پرانتز باز و بسته که اسم ستون در آن ذکر می شود، اسم کلید اصلی را درج می کنیم. فرمول:

```
CONSTRAINT PrimaryKeyName PRIMARY KEY(ColumnName)
```

در این فرمول، کلیدواژه ی CONSTRAINT و عبارت PRIMARY KEY الزامی هستند. در جایگاه PrimaryKeyName، اسمی که می خواهید به کلید اصلی تخصیص دهید را ذکر کندی. داخل پرانتزهای عبارت PRIMARY KEY، اسم ستونی که به عنوان کلید اصلی بکار می رود را درج نمایید. مثال:

```
CREATE TABLE Persons
(
   PersonID int NOT NULL,
   FirstName nvarchar(20),
   LastName nvarchar(20) NOT NULL,
   CONSTRAINT PrimKeyPeople PRIMARY KEY(PersonID)
);
GO
```

رایج است که اسم PRIMARY KEY از مخفف آن (PK) و اسم جدول تشکیل شود. مثال:

```
CREATE TABLE Persons
(
   PersonID int NOT NULL,
   FirstName nvarchar(20),
   LastName nvarchar(20) NOT NULL,
   CONSTRAINT PK_Persons PRIMARY KEY(PersonID)
);
GO
```

آموزش کاربردی: ایجاد کلید اصلی

1. تمامی دستورات داخل پنجره ی Query Editor را با گرفتن کلیدهای Ctrl + A انتخاب نمایید.

2. برای ایجاد تعدادی جدول که Primary key دارند، دستورات زیر را وارد نمایید:

```
USE RealEstate1;
```

```
GO
CREATE TABLE Listing.PropertiesTypes
(
   PropertyType nvarchar(20) PRIMARY KEY not null,
   [Description] nvarchar(max),
);
GO
CREATE TABLE Listing.PropertiesConditions
(
   Condition nvarchar(20),
   [Description] nvarchar(max),
   CONSTRAINT PK_PropertiesConditions PRIMARY KEY(Condition)
);
GO
CREATE TABLE Listing.SalesStatus
(
   SaleStatus nvarchar(20),
   [Description] nvarchar(max),
   CONSTRAINT PK_SalesStatus PRIMARY KEY(SaleStatus)
);
GO
```

3. برای ایجاد پایگاه داده، در منوی اصلی: Query -> Execute.

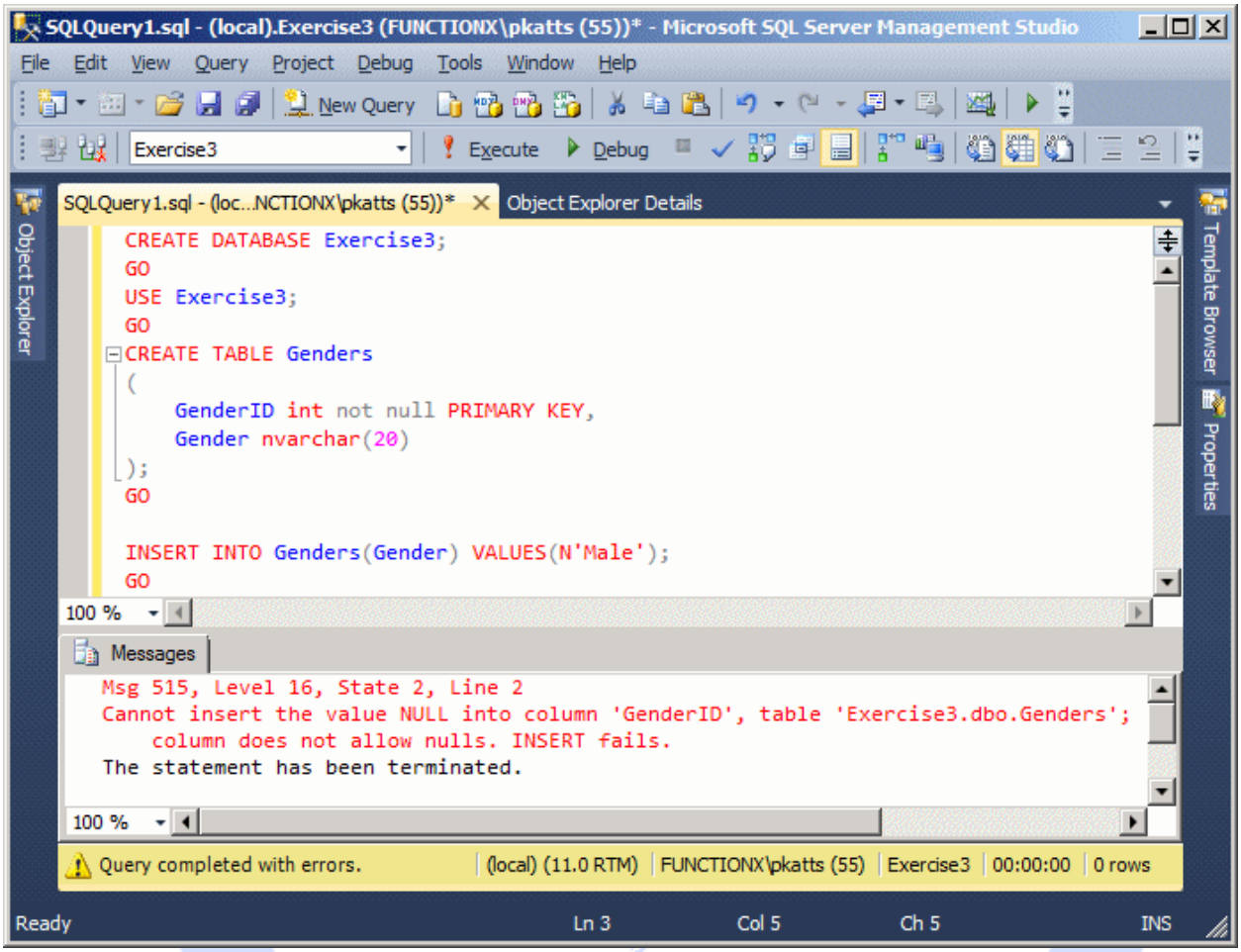
رکورد، درج داده و کلید اصلی

ستونی که کلید اصلی یک جدول انتخاب می شود، بایستی درون تمامی فیلدهای خود مقدار داشته باشد (مقدار null در این ستونی جایی ندارد). بنابراین null و رشته ی خالی در فیلدهای این ستون پذیرفته نیستند. در صورتی که حتی یکی از فیلدهای ستون تهی باشد و مقداری در آن درج نکنید، خطا صادر می شود. مثال:

```
CREATE DATABASE Exercise3;
GO
USE Exercise3;
GO
CREATE TABLE Genders
(
   GenderID int not null PRIMARY KEY,
   Gender nvarchar(20)
);
GO

INSERT INTO Genders(Gender) VALUES(N'Male');
GO
```

نتیجه ی زیر را برمی گرداند:



اولین نکته ای که باید به آن توجه داشته باشید، نوع داده ای ستون کلید اصلی می باشد.

آموزش کاربردی

1. با گرفتن کلیدهای Ctrl + A، تمامی دستورات را داخل Query Editor انتخاب نمایید.

2. برای ایجاد تعدادی سطر برای جدول و وارد کردن مقادیر برای فیلدهای ستون کلید اصلی، دستور زیر را وارد نمایید:

```
USE RealEstate1;
GO
INSERT INTO Listing.PropertiesTypes
VALUES(N'Condominium', N'A condominium, also called condo, is a unit built in a small, medium, or large building. It ressembles an apartment. It may have one, two, or more bedrooms.'),
    (N'Townhouse', N'A townhouse, sometimes called a town house or town home, is a relatively small house attached to at least another house.'),
    (N'Single Family', N'A single family is a stand-alone house. It may have one, two or three levels, also called stories.');
GO
INSERT INTO Listing.PropertiesConditions
VALUES(N'Excellent', N'An excellent property is one that has everything perfect or almost. There are no major repairs to be made.'),
    (N'Good', N'A property is good if it is good enough to be sold. It may be less than perfect but it is wholly acceptable.'),
```

```
    (N'Needs Repair', N'This type of condition indicates that one or more repairs are necessairy. The property in this condition is
not ready for sale.'),
    (N'Bad Shape', N'A property is in bad shape if it requires a mojor or many repairs.');
GO
INSERT Listing.SalesStatus
VALUES(N'Ready For Sale', N'The property is currently available for sale.'),
    (N'Sold', N'The property has been sold.');
GO
```

3. برای ایجاد پایگاه داده، در منوی اصلی: Query -> Execute.

نوع مقدار کلید اصلی

نوع داده ای مقداری که در فیلدهای ستون کلید اصلی می نشیند اغلب int یا عدد صحیح می باشد. برای ایجاد چنین ستونی، کافی است نوع داده ای integer را به آن تخصیص دهید. اگر دقیقا محدوده ی مقادیری که در ستون کلید اصلی جای می گیرند را می دانید، در آن صورت می توانید نوع داده ای tinyint یا small int را به آن ستون تخصیص داد. مثال:

```
USE Exercise1;
GO
CREATE TABLE Personnel.Genders
(
        GdrNbr tinyint,
        Gender nvarchar(20),
        CONSTRAINT PK_Genders PRIMARY KEY(GdrNbr)
);
GO
CREATE TABLE Personnel.MaritalsStatus
(
        StatusCode smallint,
        MaritalStatus nvarchar(32),
        CONSTRAINT PK_MaritalStatus PRIMARY KEY(StatusCode)
);
GO
```

در زمان درج داده در فیلدی که نوع کلید اصلی آن عدد صحیح (integer) می باشد، می بایست مقدار دلخواه را ارائه کرده در عین حال اطمینان حاصل نمایید آن مقدار در میان دیگر مقادیر همان ستون منحصر بفرد و غیر تکراری باشد.

علاوه بر integer، نوع داده ای رشته (string) از دیگر انواع داده ای پرکاربرد و استفاده برای کلید اصلی می باشد. برای این منظور می توانید char، varchar و مشتقات آن ها را بکار ببرید. توجه: تحت هیچ شرایطی نباید نوع داده ای text را به عنوان نوع کلید اصلی انتخاب کنید. در زمان درج داده، مطمئن شوید مقدار مورد نظر (از نوع رشته) داخل تک کوتیشن محصور شده، سپس برای هر رکورد ارائه می شود.

اصول و قوانین پیشنهادی برای تعریف کلیدهای اصلی

تعدادی اصل و دستور العمل وجود دارد که در زمان ایجاد کلید اصلی باید به آن توجه داشته باشید:

1. بایستی از انتخاب مقادیری که احتمال تغییر آن ها وجود دارد (به عنوان کلید اصلی) خودداری نمایید. به عنوان مثال تمامی شهروند های ایالات متحده یک شماره ی بیمه منحصربفرد دارند، اما می توانند آن را در صورت نیاز تغییر دهند.

2. از بکار بردن اسامی خاص به عنوان کلید اصلی اجتناب کنید. گرچه لزومی به گفتن آن وجود ندارد، با این حال برخی کاربران این اشتباه را مرتکب می شوند. به عنوان نمونه، هیچ تضمینی وجود ندارد که ترکیبی از دو اسم منحصر بفرد باشد (برای مثال ممکن است اشخاص متعددی با اسم Michael Jordans یا Michael Jacksons وجود داشته باشند).

3. نباید نوع داده ای decimal و مشتقات آن را به عنوان کلید اصلی یک جدول بکار ببرید. به این خاطر که مقدار کلید اصلی باید ثابت و دقیق باشد، در حالی که مقادیر اعشاری متغیر و غیر قابل پیشبینی هستند.

4. همچنین نوع داده ای date و time (از آنجایی که نامشخص و غیر قطعی هستند) کاندیدهای مناسبی برای کلید اصلی محسوب نمی شوند.

قید کلید خارجی (Foreign Key Constraint)

کار خود را با پایگاه داده ی بانک ادامه می دهیم. تصور کنید یک مشتری به بانک مراجعه نموده تا پولی را در حساب خود واریز کند. همان طور که قبلا گفته شد، لزومی ندارد هر بار که مشتری قصد انجام تراکنش جدیدی را دارد، حساب جدیدی ایجاد گردد. بلکه اطلاعات مشتری از وی دریافت شده و در جدولی ذخیره می گردد و بعد به وسیله این جدول تراکنش های مشتری پردازش می شود. همان طور که پیش تر تشریح شد، جدول account باید اطلاعات خود را در اختیار دیگر جداول پایگاه داده که به آن اطلاعات نیاز دارند، قرار دهد. برای اینکه گردش اطلاعات از جدول به جدول دیگر امکان پذیر شود، باید یک رابطه بین آن ها ایجاد کرد.

برای اینکه جدول B بتواند اطلاعات جدول A را داشته باشد، B بایستی ستونی داشته باشد که مقادیر جدول A را به نمایندگی از آن جدول در خود ذخیره کند. این ستون در واقع به مثابه ی یک سفیر یا پیوند ایفای نقش می کند. ستون ذکر شده در اصل به جدول B تعلق ندارد بلکه صرفا به دو جدول A و B این امکان را می دهد که با هم ارتباط داشته باشند. به این دلیل، ستونی که در اصل به A تعلق دارد اما در جدول B نیز لحاظ شده است، ستون کلید خارجی یا foreign key اطلاق می گردد.

کلید خارجی یک ستون است که اطلاعات آن متعلق به جدول دیگر است. برخلاف کلید اصلی که باید یکتا و غیرتکراری باشد، یک جدول می تواند بی نهایت کلید خارجی داشته باشد، زیرا هر کلید خارجی اساسا به جدول دیگری تعلق دارد (مربوط می باشد). Microsoft، حداکثر تعداد 253 کلید خارجی را به ازای هر جدول اجازه می دهد.

ایجاد یک کلید خارجی از طریق پنجره ی طراحی جدول (Table Design View)

برای ایجاد یک کلید خارجی به صورت ویزاردی (از طریق پنجره ی Table Design)، داخلی جدولی که کلید خارجی را دریافت می کند، کافی است یک ستون با پیروی از قوانین زیر ایجاد نمایید:

1. توصیه می شود اسم ستون با اسم کلید اصلی جدول یکسان باشد (این صرفا یک پیشنهاد بوده و از ملزومات نمی باشد).

2. ستون می بایست از نظر نوع داده ای با ستون کلید اصلی جدول یکسان باشد.

طبیعتا برای اینکه اطلاعات از یک جدول به جدول دیگر جریان داشته باشد، یک ستون حاوی اطلاعات کلید اصلی باید در جدول پدر ایجاد شده باشد. این ستون را می توان قبل یا بعد از ساخت جدول ایجاد نمود. بنابراین مادام اینکه از پیش هیچ پیوندی بین دو جدول ایجاد نشده باشد، ترتیب ایجاد آن ها اهمیتی ندارد.

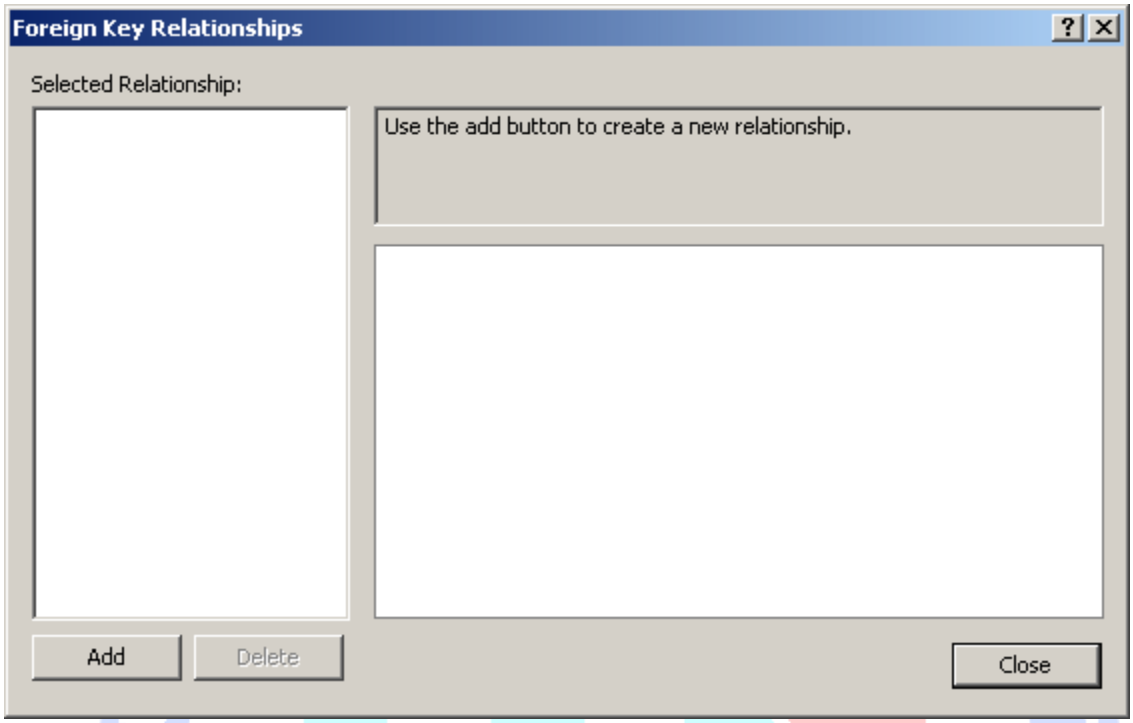
جدول دربردارنده ی کلید اصلی که دیگر جداول از اطلاعات آن استفاده می کنند، در اصطلاح جدول اصلی یا پدر خوانده می شود و جدولی که اطلاعات شی (جدول پدر) ذکر شده را دریافت می کند، جدول خارجی یا فرزند نامیده می شود.

ایجاد یک کلید خارجی در کادر محاوره ای Relationships

به منظور ایجاد کلید خارجی در یک جدول:

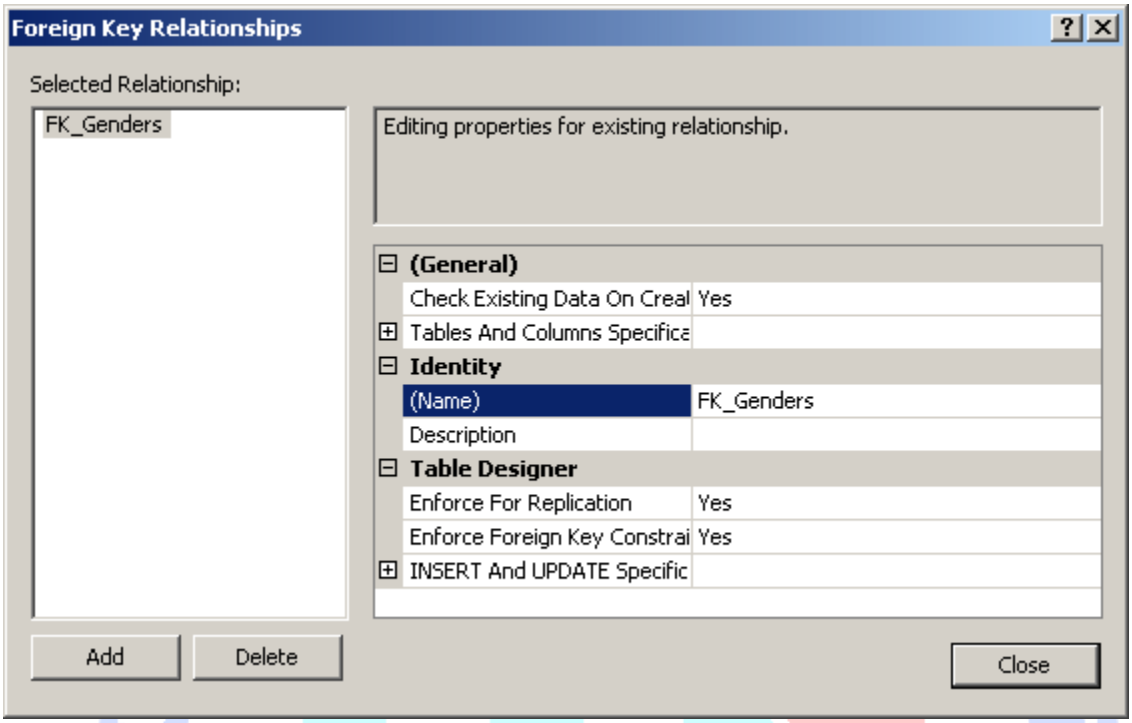
1. به Object Explorer مراجعه نموده و جدول فرزند را در Design View باز کنید.

2. داخل جدول راست کلیک کرده و گزینه ی ...Relationships را انتخاب نمایید.



3. در کادر محاوره ای Foreign Key Relationships، دکمه ی Add را کلیک نمایید.

4. یک اسم پیش فرض به شما پیشنهاد می شود. می توانید آن را پذیرفته یا تغییر دهید. برای تغییر اسم کلید خارجی، در سمت راست Identity را باز کرده و رشته ی مورد نظر را در فیلد Name تغییر دهید:



5. در صورت لزوم، می توانید با زدن دکمه ی Add کلیدهای خارجی دیگر ایجاد کنید. برای حذف کردن کلید خارجی، می بایست ابتدا آن را در کادر Selected Relationships انتخاب کرده و سپس دکمه ی Delete را فشار دهید.

ایجاد یک کلید خارجی با دستوارت SQL

می توانید کلید خارجی را با استفاده از دستورات SQL و با کدنویسی ایجاد نمایید. دستور نحوی آن به شرح زیر می باشد:

```
FOREIGN KEY REFERENCES ParentTableName(ForeignKeyCcolumn)
```

عبارت FOREIGN KEY و کلیدواژه ی REFERENCES هر دو الزامی می باشند. در جایگاه ParentTableName، اسم جدول پدر دربردارنده ی اطلاعاتی که در جدول جاری لحاظ شده و مورد دسترسی قرار می گیرد را درج نمایید. داخل پرانتزهای ParentTableName، اسم ستون کلید اصلی جدول پدر را درج نمایید. مثال:

```
CREATE TABLE Persons
(
   PersonID int PRIMARY KEY NOT NULL,
   FirstName nvarchar(20),
   LastName nvarchar(20) NOT NULL,
   GenderID int NULL FOREIGN KEY REFERENCES Genders(GenderID)
);
GO
```

قید (constraint) بر روی کلید خارجی

همان طور که در کد بالا مشاهده می کنید، هیچ اسمی به کلید خارجی اختصاص نیافته است. اگر اسمی برای کلید خارجی درنظر نگیرید، مفسر SQL به صورت خودکار یک اسم پیش فرض برای کلید خارجی مشخص می کند. برای تخصیص اسم دلخواه به کلید خارجی، پس از ایجاد ستون مورد نظر، کلیدواژه ی CONSTRAINT و به دنبال آن نام مد نظر را تایپ کنید. سپس باقی دستور را همان طور که بالا نمایش داده شد، ادامه دهید. مثال:

```
CREATE TABLE Persons
(
   PersonID int PRIMARY KEY NOT NULL,
   FirstName varchar(20),
   LastName varchar(20) NOT NULL,
   GenderID int NULL CONSTRAINT FKGenders
            FOREIGN KEY REFERENCES Genders(GenderID)
);
GO
```

می توان برای نام گذاری ستون کلید خارجی لزوما از قوانین SQL پیروی نکنیم و خود یک اسم برای آن تنظیم کنیم. برای تخصص اسم دلخواه به یک کلید خارجی، آن را باید به عنوان یک قید تعریف نمایید. برای این منظور، پس از تعریف ستون حاوی کلید خارجی و پیش از پرانتز بسته، یک کلید خارجی با پیروی از فرمول زیر ایجاد نمایید:

```
CONSTRAINT Name FOREIGN KEY(Foreign Key) REFERENCES Parent(Foreign Key)
```

مثال:

```
CREATE TABLE Genders
(
   GenderID int NOT NULL,
   Gender nvarchar(20) NOT NULL,
   CONSTRAINT PK_Genders PRIMARY KEY(GenderID)
);
GO

CREATE TABLE Persons
(
   PersonID int PRIMARY KEY NOT NULL,
   FirstName nvarchar(20),
   LastName nvarchar(20) NOT NULL,
   GenderID int,
   Comments nvarchar(max),
   CONSTRAINT FK_Genders FOREIGN KEY(GenderID) REFERENCES Genders(GenderID)
);
GO
```

آموزش کاربردی: ایجاد کلید خارجی

1. در محیط Query Editor کلیک کرده و تمامی دستورات داخل آن را انتخاب نمایید.

2. برای ایجاد یک جدول که دارای کلید خارجی (foreign key) می باشد، کد زیر را وارد نمایید:

```
USE RealEstate1;
```

```
GO
CREATE TABLE Listing.Properties
(
   PropertyNumber nvarchar(12),
   [Address] nvarchar(60),
   City nvarchar(50),
   [State] nchar(2),
   ZIPCode nvarchar(12),
   PropertyType nvarchar(20) FOREIGN KEY(PropertyType) REFERENCES Listing.PropertiesTypes(PropertyType),
   Bedrooms smallint,
   Bathrooms float,
   Stories smallint,
   FinishedBasement bit,
   IndoorGarage tinyint,
   YearBuilt smallint,
   Condition nvarchar(20) CONSTRAINT FK_PropertiesConditions FOREIGN KEY REFERENCES
Listing.PropertiesConditions(Condition),
   MarketValue money,
   SaleStatus nvarchar(20),
   CONSTRAINT PK_Properties PRIMARY KEY(PropertyNumber),
   CONSTRAINT FK_SalesStatus FOREIGN KEY(SaleStatus) REFERENCES Listing.SalesStatus(SaleStatus)
);
GO
```

3. جهت ایجاد پایگاه داده ی جدید، در منوی اصلی: Query -> Execute.

ایجاد رابطه بین جداول

همان طور که قبلا تشریح شد، پایگاه داده ی رابطه ای یک سیستم است که در آن اطلاعات از یک جدول به جدول دیگر جریان دارد. برای فراهم آوردن این امکان (و آماده سازی جداول) باید کلیدهای اصلی و خارجی ایجاد می کنیم. پس از آماده سازی جدول، بایستی آن ها را به هم لینک کنید که از آن به عنوان ایجاد رابطه بین جداول نیز یاد می شود.

اگر کلید خارجی را با کدنویسی ایجاد نکرده اید، در آن صورت می بایست رابطه را به هنگام ایجاد رابطه بین دو جدول ایجاد کنید.

نحوه ی ایجاد رابطه بین دو جدول

1. جدول فرزند را داخل design view باز نمایید.

2. داخل جدول راست کلیک نموده و گزینه ی ...Relationships را انتخاب کنید. اگر کلید خارجی تعریف نشده، بر روی دکمه ی Add کلیک کرده و اسم آن را در فیلد Name، زیر Identity وارد نمایید.

3. در کادر Selected Relationships، بر روی کلید خارجی که رابطه را برقرار می کند کلیک کنید.

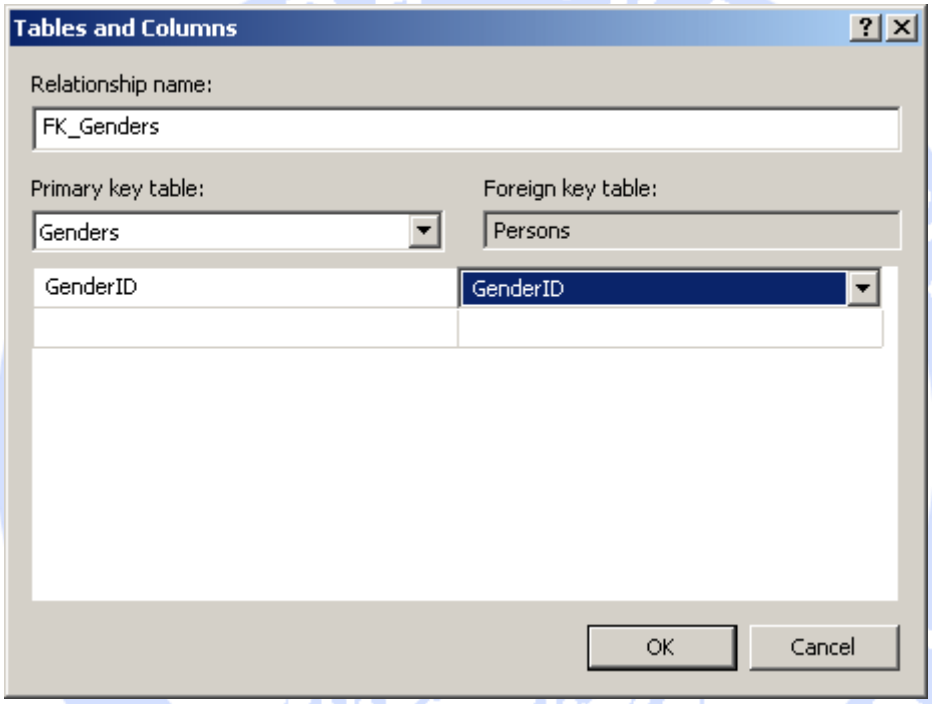
4. در سمت راست کادر یاد شده، گره Tables And Columns Specification را گشوده، سپس بر روی دکمه ی نقطه چین ... کلیک نمایید.

5. در لیست کشویی Primary Key Table، جدول پدر که دربردارنده ی کلید اصلی است را انتخاب نمایید.

6. در زیر جدول پدر، بر روی ستون کلید اصلی کلیک نموده و آن را انتخاب نمایید.

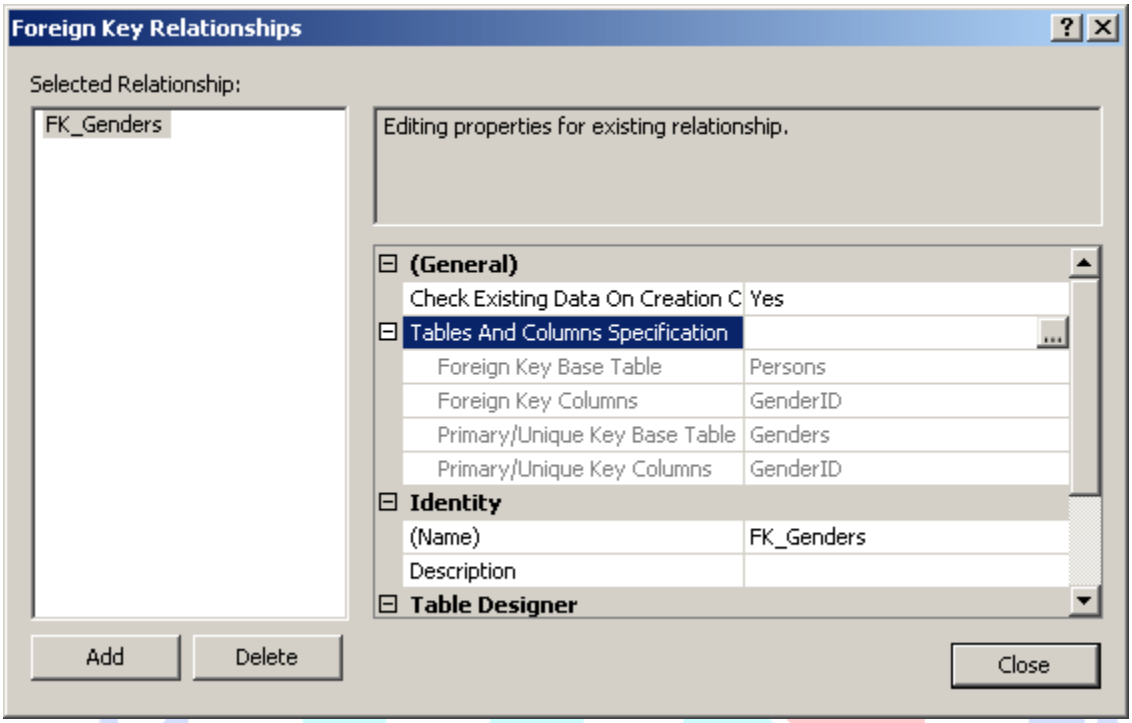
7. در زیر Foreign Key Table، اسم جدول جاری را مشخص نمایید.

8. در زیر اسم جدول فرزند، اسم ستون کلید خارجی را انتخاب نمایید. مثال:



9. OK را کلیک کنید.

10. زمانی که یک رابطه بین دو جدول ایجاد شد، این رابطه در بخش Tables And Column Specification قابل مشاهده می باشد.

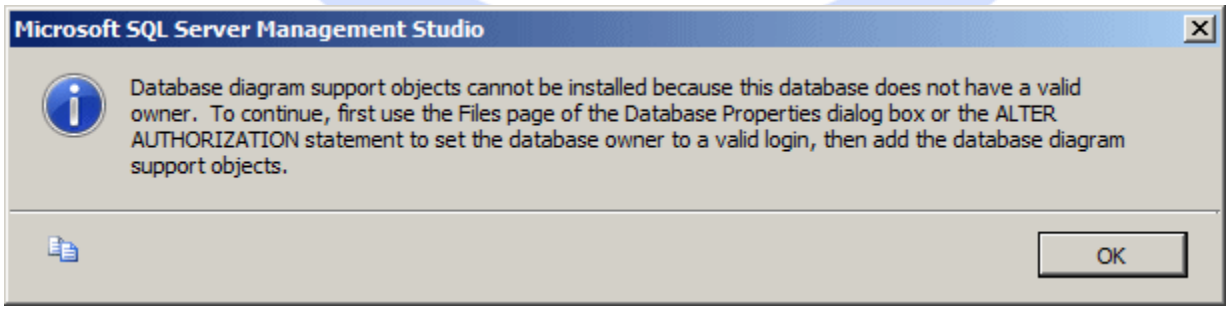


11. به همین نحو می توان با کلیک بر روی Add و پیکرنبدی لینک روابط متعدد دیگر ایجاد کرد.

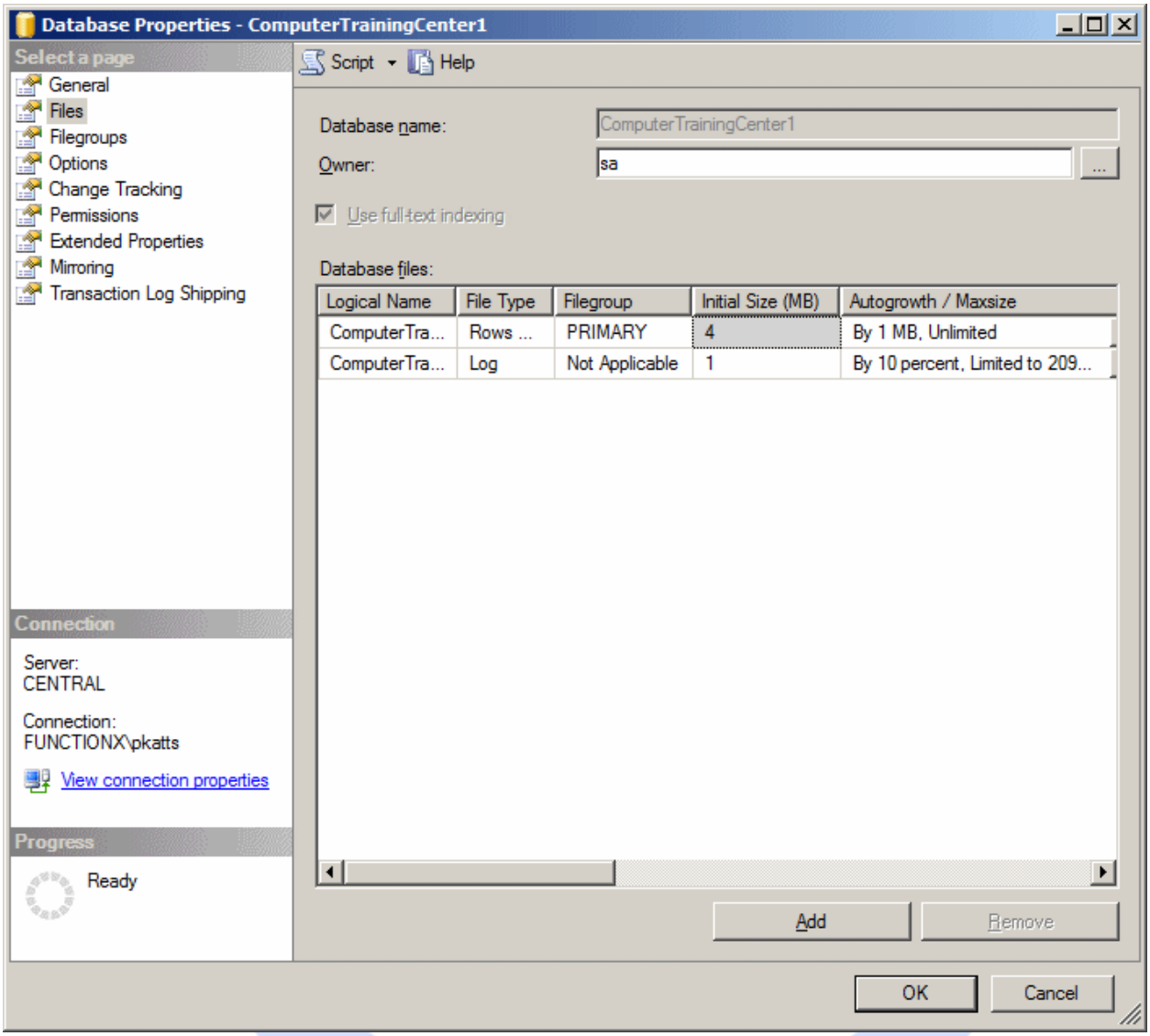
پنجره ی Diagrams

Diagram یک پنجره است که روابط بین جداول یک پایگاه داده را به نمایش می گذارد. برای ایجاد یک diagram:

1. در پنجره ی Object Explorer، داخل گره database، بر روی Database Diagrams کلیک نمایید. ممکن است یک کادر حاوی این خطا نمایش داده شود، پایگاه داده ی شما مالک مشخصی ندارد:

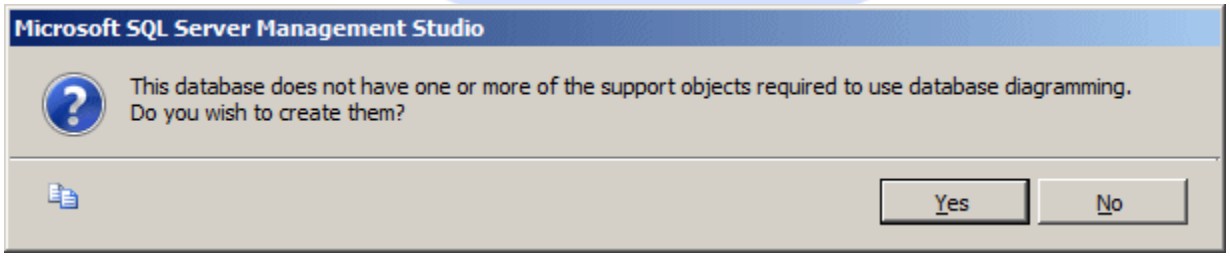


در صورت دریافت این خطا، دکمه ی OK را کلیک نمایید. حال بر روی اسم پایگاه داده راست کلیک کرده و سپس گزینه ی Properties را انتخاب نمایید. در لیست سمت چپ، Files را انتخاب کنید. در تکس باکس Owner، واژه ی sa را تایپ نمایید:



دکمه ی OK را کلیک نمایید. بار دیگر Database Diagrams را کلیک کنید.

2. یک کادر محاوره ای به شما اعلان می کند که پایگاه داده فعلی diagram ندارد.



پیغام را خوانده و دکمه ی Yes را کلیک نمایید.

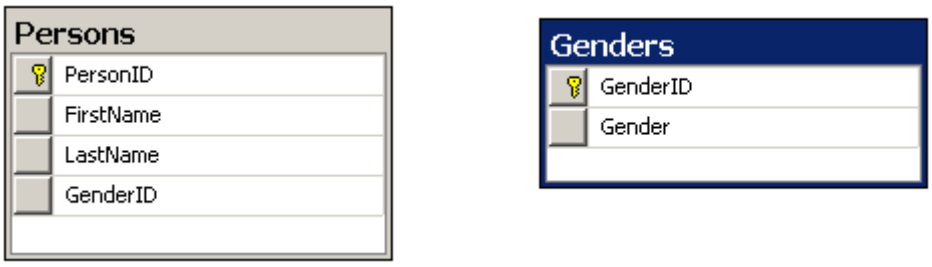
3. بر روی Database Diagrams راست کلیک کرده و سپس گزینه ی New Database Diagram را انتخاب نمایید.

4. در کادر محاوره ای Add Table، بر روی تک تک جدول ها کلیک کرده و سپس دکمه ی Add را فشار دهید. همچنین می توانید بر روی جدول دابل کلیک کرده تا یک جدول جدید اضافه شود.

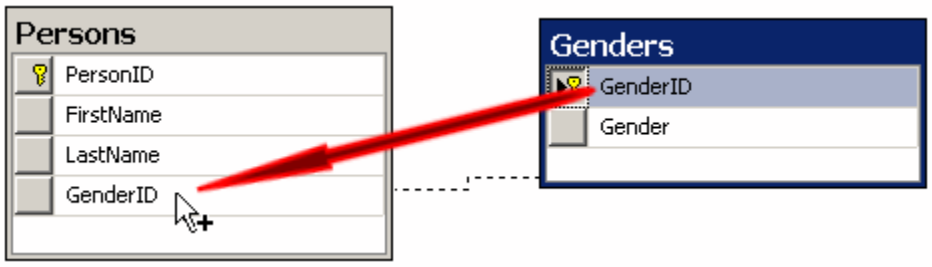
5. در کادر محاوره ای Add Table، بر روی دکمه ی Close کلیک نمایید.

در نوار ابزار، بر روی دکمه ی Zoom کلیک کرده و سپس یک مقدار بزرگتر یا کوچکتر انتخاب نمایید.

برای جابجایی جدول، می توانید بر روی نوار عنوان آن کلیک کرده و آن را بکشید. مثال:



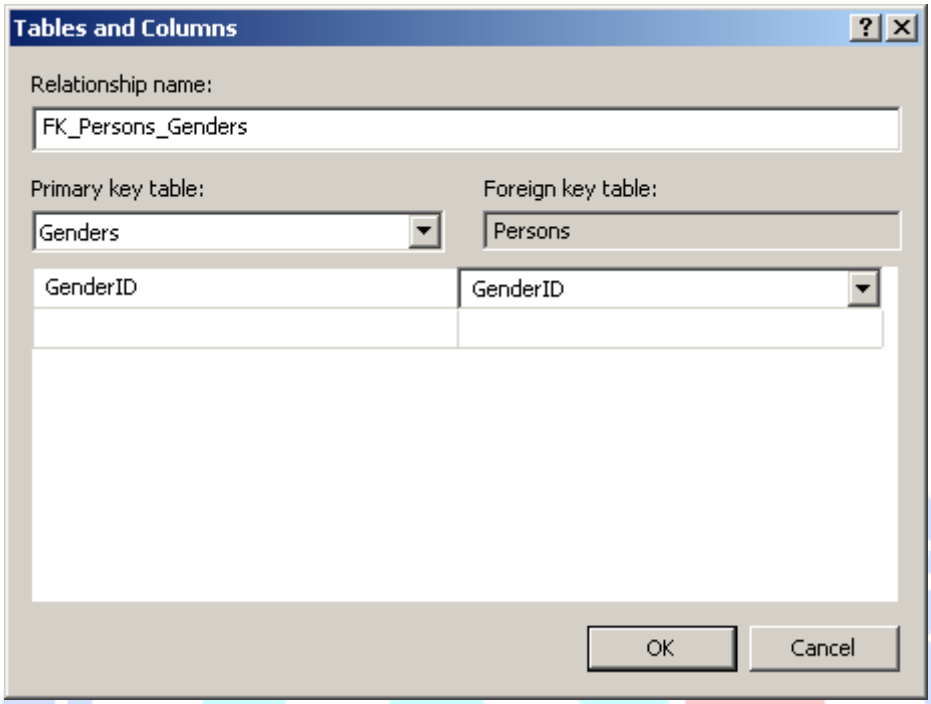
6. برای ایجاد رابطه، بر روی کادر خاکستری سمت چپ هر ستون در جدول پدر کلیک کرده و آن را بکشید، سپس آن را بر روی ستون موجود در جدول فرزند رها (drop) کنید. روش بهتر این است که بر روی کادر خاکستری ستون کلید اصلی (در جدول پدر) کلیک کرده، آن کادر را بکشید و سپس در ستون کلید خارجی جدول فرزند جای گذاری کنید:



7. کادر محاوره ای Tables and Columns ظاهر می شود. این کادر ستونی که کشیده شده و نیز ستونی که ستون کلید اصلی بر روی آن رها شده است را نمایش می دهد.

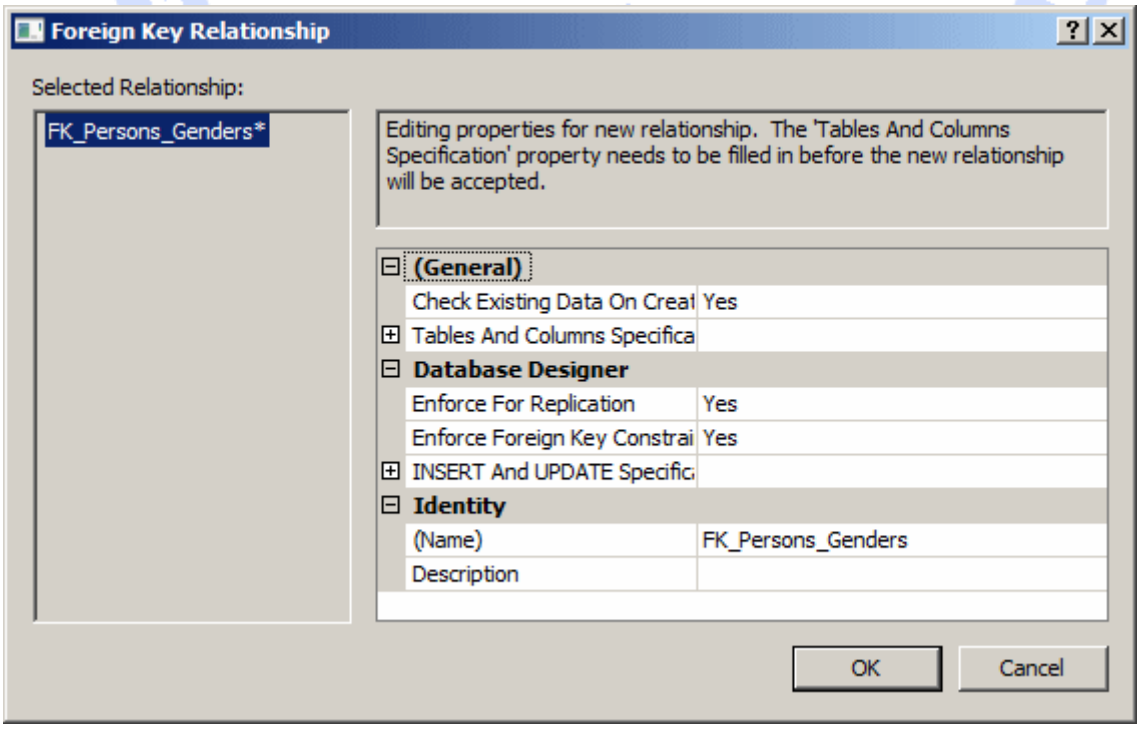
8. در زیر Primary Key Table، جدول پدر ذکر می شود. در زیر جدول پدر، یک لیست کشویی مشاهده می کنید که می توانید در آن ستون کلید اصلی را مشخص نمایید.

9. در زیر کادر Foreign key table، ستون کلید خارجی را در جدول فرزند مشخص نمایید.

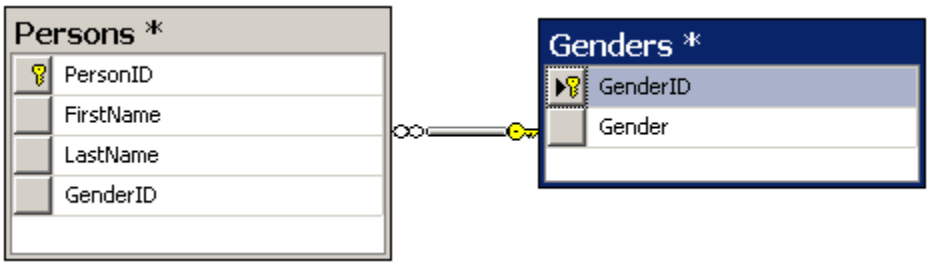


10. بر روی دکمه OK کلیک کنید.

11. کادر محاوره ای دیگری با عنوان Foreign Key Relationship پدیدار می شود. این کادر به شما اجازه می دهد رابطه را مدیریت نمایید:

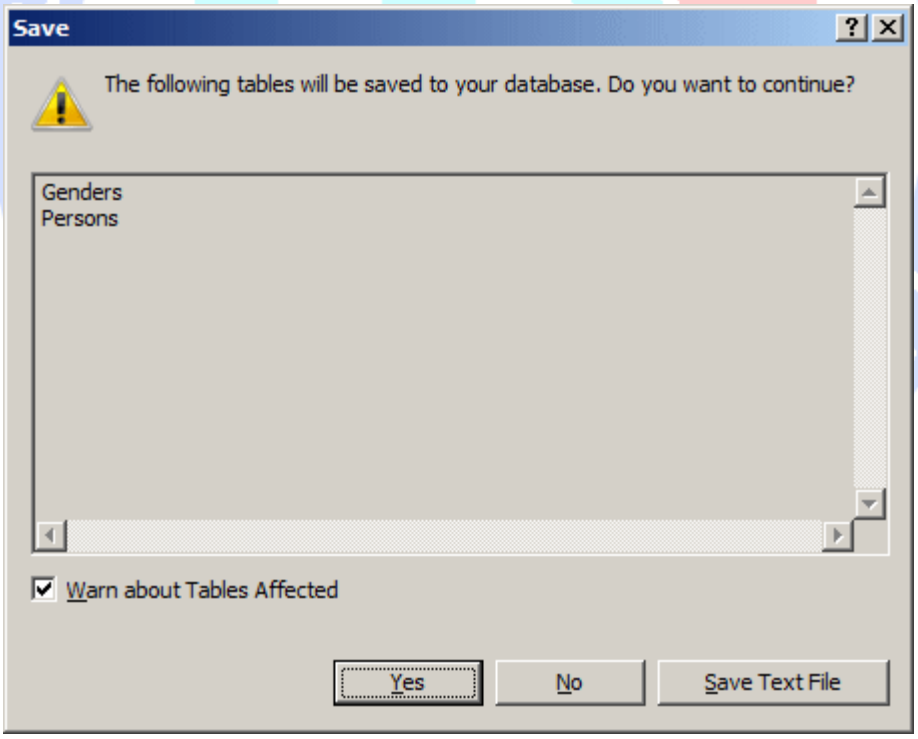


12. پس از خواندن و اجرای عملیات لازم در این کادر محاوره ای، بر روی OK کلیک نمایید. در پی این این کار لینکی بین جداول ایجاد می شود.



13. به همین طریق می توان چندین رابطه ی دیگر ایجاد کرد. پس از انجام عملیات لازم، تغییرات را ذخیره نموده و پنجره ی diagram را ببندید.

14. یک پنجره ی محاوره ای به شما اطلاع می دهد که جداول متصل به هم (جداولی که توسط کلید اصلی و خارجی با هم رابطه پیدا کردند)، باید ذخیره شوند.



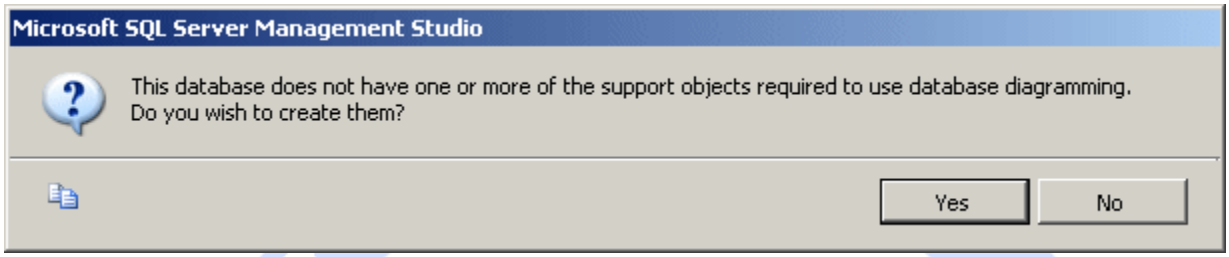
15. بر روی دکمه ی Yes کلیک نمایید.

نحوه ی ایجاد diagram

1. در پنجره ی Object Explorer، بر روی Databases راست کلیک کرده و گزینه ی Refresh را انتخاب نمایید.

2. گره AltairRealtors را باز نمایید و سپس بر روی گره Database Diagrams آن کلیک کنید.

3. حال اگر بر روی آن کلیک کنید، کادر محاوره ای ظاهر شده و به شما پیغام می دهد که این پایگاه داده فاقد diagram می باشد.



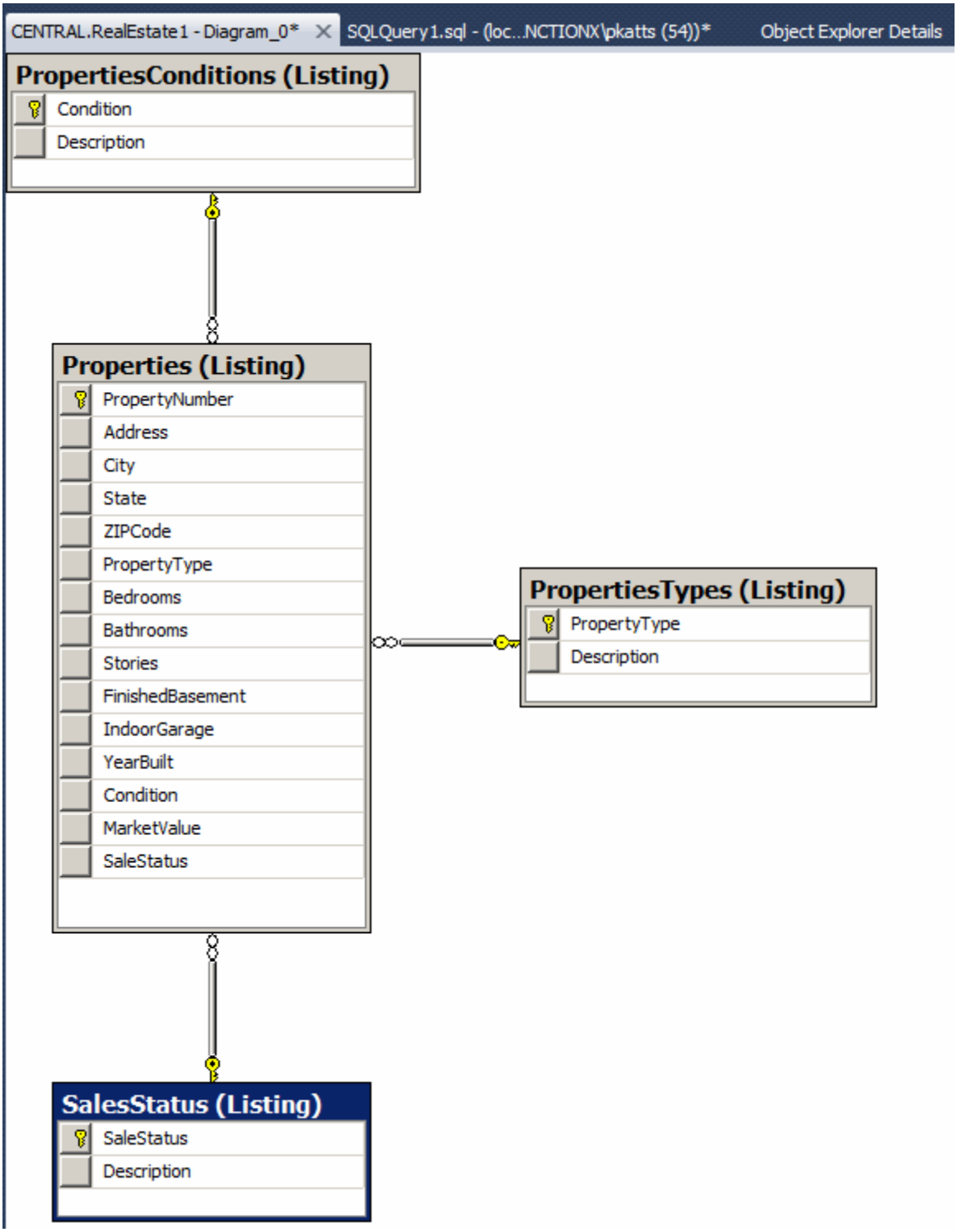
4. پس از خواندن محتویات آن دکمه ی Yes را کلیک نمایید.

5. بر روی Database Diagrams راست کلیک نموده، سپس گزینه ی New Database Diagram را انتخاب نمایید.

6. در کادر محاوره ای Add Table، بر روی PropertiesTypes (Listing) کلیک نمایید و سپس دکمه ی Add را فشار دهید.

7. بر روی PropertiesConditions (Listing) دابل کلیک نموده تا اضافه شود.

8. در کادر محاوره ای Add Table، بر روی Properties (Listing) و SalesStatus (listing) دابل کلیک کرده، سپس دکمه ی Close را کلیک نمایید. همان طور مشاهده می کنید، رابطه های لازم بین جداول ایجاد شده است:



9. برای ذخیره اطلاعات diagram (نمودار) جاری، در Standard toolbar دکمه ی Save را کلیک نمایید.

10. اسم نمودار را dgmAltairRealtors را انتخاب نموده و OK را کلیک نمایید.

11. پنجره ی diagram را ببندید.

درج اطلاعات در فیلد کلید اصلی

همان طور که پیش تر ذکر شد، در زمان ایجاد رکوردهای یک جدول، باید مقدار مناسب را به فیلد کلید اصلی تخصیص دهید. هر فیلد داخل ستونی که کلید اصلی انتخاب شده باید دارای مقدار منحصر بفرد باشد.

درج اطلاعات در فیلد کلید خارجی

مقدار ستون کلید خارجی باید در ستون (فیلد) کلید اصلی جدول پدر موجود باشد. اگر مقدار در جدول پدر یافت نشد، در آن صورت خطا صادر می شود. اگر ملزوم به استفاده از مقداری در جدول فرزند هستید که در جدول پدر وجود ندارد، بایستی ابتدا آن را در جدول پدر ایجاد نمایید سپس به سراغ جدول فرزند رفته و آن رکورد را ایجاد نمایید.

ایجاد مقادیر کلید خارجی

1. کلیه ی دستورات موجود در Query Editor را با زدن کلید های Ctrl + A انتخاب نمایید.

2. برای ایجاد تعدادی رکورد و افزودن مقادیری در فیلدهای ستون کلید خارجی، کد زیر را در پنجره ی Query Editor وارد نمایید:

```
USE RealEstate1;
GO
INSERT INTO Listing.Properties
VALUES(N'52408180', N'1640 Lombardo Ave', N'Silver Spring', N'MD', N'20904', N'Single Family', 4, 2.5, 3, 1, 2, 1995,
N'Good', 495880.00, N'Ready For Sale'),
    (N'68830614', N'10315 North Hacht Rd', N'College Park', N'MD', N'20747', N'Single Family', 4, 3.5, 3, 1, 2, 2000,
N'Excellent', 620724.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State],
 ZIPCode, Condition, FinishedBasement, Stories, MarketValue, SaleStatus)
VALUES(N'96114604', N'6366 Lolita Drive', N'Laurel', N'MD', N'20707', N'Good', 1, 2, 422625.00, N'Sold');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, PropertyType, Bedrooms, IndoorGarage, MarketValue)
VALUES(N'39485797', N'9002 Palasko Hwy', N'Tysons Corner', N'Condominium', 2, 2, 422895.00);
GO
INSERT INTO Listing.Properties(PropertyNumber, [State], ZIPCode, Bedrooms, YearBuilt, MarketValue, SaleStatus)
VALUES(N'42081159', N'DC', N'20011', 2, 1982, 312555, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, ZIPCode, PropertyType, Bedrooms, YearBuilt, MarketValue,
SaleStatus)
VALUES(N'97172037', N'Alexandria', N'22024', N'Single Family', 3, 1965, 345660.00, N'Sold');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State], PropertyType, Condition, Bedrooms, Bathrooms,
MarketValue)
VALUES(N'20880417', N'4140 Holisto Crt', N'Germantown', N'MD', N'Condominium', N'Excellent', 2, 1, 215495.00);
GO
INSERT INTO Listing.Properties
```

```
VALUES(N'92747400', N'9522 Lockwood Rd', N'Chevy Chase', N'MD', N'20852', N'Townhouse', 3, 2.5, 3, 0, 1,  1992, N'Bad
Shape', 415665.00, N'Ready For Sale'),
    (N'20708150', N'14250 Parkdoll Rd', N'Rockville', N'MD', N'20854', N'Townhouse', 3, 2.5, 2, 1, 0, 1988, N'Good',
325995.00, N'Sold');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, PropertyType, Bedrooms, YearBuilt, MarketValue)
VALUES(N'29304857', N'Washington', N'Townhouse', 4, 1975, 366775.00);
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State],
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms, YearBuilt, MarketValue)
VALUES(N'28085040', N'10340 Helmes Street #408', N'Silver Spring', N'MD', N'20906', N'Condominium', N'Good', 1, 1, 2000,
242775.00);
GO
INSERT INTO Listing.Properties
VALUES(N'24747299', N'1008 Coppen Street', N'Silver Spring', N'MD', N'20906', N'Single Family', 3, 3, 3, 1, 3, 1996,
N'Excellent', 625450.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, ZIPCode, PropertyType, Stories, YearBuilt, MarketValue)
VALUES(N'39485070', N'Chevy Chase', N'20956', N'Single Family', 3, 2001, 525450.00);
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State], PropertyType, Condition, Bedrooms, MarketValue,
SaleStatus)
VALUES(N'29380597', N'686 Herod Ave #D04', N'Takoma Park', N'MD', N'Condominium', N'Excellent', 2, 360885.00,
N'Sold');
GO
INSERT INTO Listing.Properties
VALUES(N'29744618', N'14005 Sniders Blvd', N'Laurel', N'MD', N'20707', N'Townhouse', 4, 1.5, 3, 1, 0, 2002, N'Needs
Repair', 412885.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, ZIPCode, Condition, Bedrooms, Stories, YearBuilt)
VALUES(N'28074085', N'Silver Spring', N'20905', N'Good', 4, 2, 1965);
GO
INSERT INTO Listing.Properties
VALUES(N'92417926', N'680 Prushia Rd NE', N'Washington', N'DC', N'20008', N'Single Family', 5, 3.5, 3, 0, 3, 2000, N'Good',
555885.00, N'Ready For Sale'),
    (N'29407906', N'14688 Parrison Street', N'College Park', N'MD', N'20742', N'Single Family', 5, 2.5, 2, 1, 2, 1995,
N'Excellent', 485995.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State],
  ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms, YearBuilt, MarketValue, SaleStatus)
VALUES(N'81115599', N'10340 Helmes Street #1012', N'Silver Spring',
    N'MD', N'20906', N'Condominium', N'Good', 1, 1, 2000, 252775.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties
VALUES(N'44759760', N'4201 Vilamar Ave', N'Hyattsville', N'MD', N'20782', N'Townhouse', 3, 2, 2, 1, 2, 1992, N'Excellent',
365880.00, N'Ready For Sale');
GO
```

3. حال برای ایجاد پایگاه داده، داخل منوی اصلی: Query -> Execute.

مقدار NULL و فرایند درج داده

مقدار NULL در (فیلدهای) ستون کلید اصلی و خارجی که باعث برقراری ارتباط بین جداول می شوند، جایی ندارد. به هنگام ایجاد رکوردهای یک جدول (جدول پدر) که دارای کلید اصلی (primary key) می باشد، اگر می دانید که جدول فرزند سطرهایی خواهد داشت که مقادیر آن در جدول پدر وجود ندارد، در آن صورت می بایست یک رکورد ساختگی یا به

اصطلاح dummy در جدول پدر ایجاد کنید که مقادیری نظیر N/A، Not Available یا Unknown را درخود نگه می دارد.

آموزش کاربردی: مدیریت مقدار null

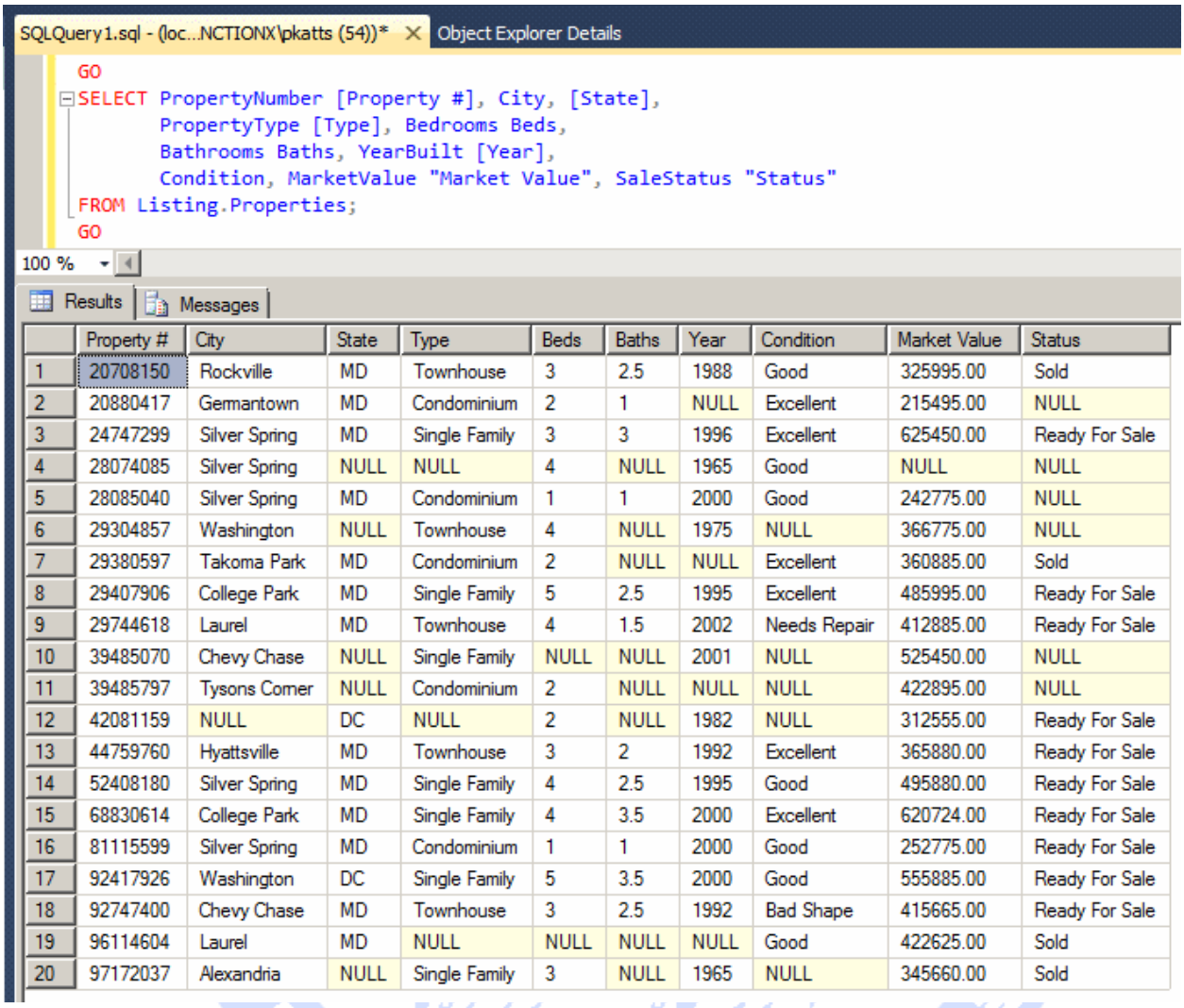
1. داخل Query Editor کلیک نموده و با زدن کلیدهای Ctrl + A به طور همزمان کد جاری را انتخاب نمایید.

2. برای نمایش فهرستی از املاک، کد زیر را تایپ نمایید:

```
USE RealEstate1;
GO
SELECT PropertyNumber [Property #], City, [State],
    PropertyType [Type], Bedrooms Beds,
    Bathrooms Baths, YearBuilt [Year],
    Condition, MarketValue "Market Value", SaleStatus "Status"
FROM Listing.Properties;
GO
```

3. برای اجرای دستور، در منوی اصلی: Query -> Execute.

4. همان طور که مشاهده می کنید، کوئری 20 رکورد از جدول مورد نظر بازیابی کرده است. همچنین تعدادی از فیلدهای ستون Status مقدار null را نشان می دهند.



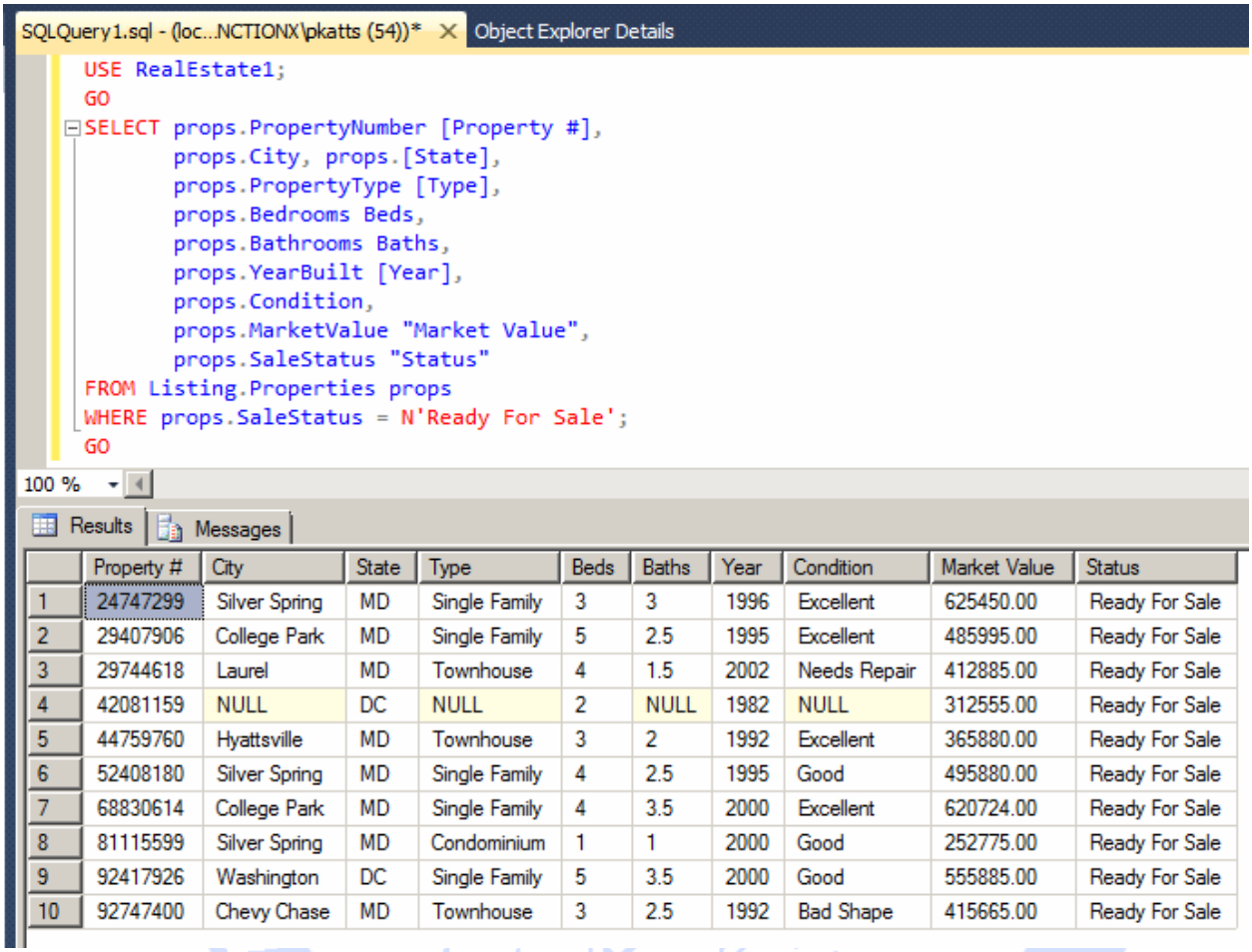
	Property #	City	State	Type	Beds	Baths	Year	Condition	Market Value	Status
1	20708150	Rockville	MD	Townhouse	3	2.5	1988	Good	325995.00	Sold
2	20880417	Germantown	MD	Condominium	2	1	NULL	Excellent	215495.00	NULL
3	24747299	Silver Spring	MD	Single Family	3	3	1996	Excellent	625450.00	Ready For Sale
4	28074085	Silver Spring	NULL	NULL	4	NULL	1965	Good	NULL	NULL
5	28085040	Silver Spring	MD	Condominium	1	1	2000	Good	242775.00	NULL
6	29304857	Washington	NULL	Townhouse	4	NULL	1975	NULL	366775.00	NULL
7	29380597	Takoma Park	MD	Condominium	2	NULL	NULL	Excellent	360885.00	Sold
8	29407906	College Park	MD	Single Family	5	2.5	1995	Excellent	485995.00	Ready For Sale
9	29744618	Laurel	MD	Townhouse	4	1.5	2002	Needs Repair	412885.00	Ready For Sale
10	39485070	Chevy Chase	NULL	Single Family	NULL	NULL	2001	NULL	525450.00	NULL
11	39485797	Tysons Corner	NULL	Condominium	2	NULL	NULL	NULL	422895.00	NULL
12	42081159	NULL	DC	NULL	2	NULL	1982	NULL	312555.00	Ready For Sale
13	44759760	Hyattsville	MD	Townhouse	3	2	1992	Excellent	365880.00	Ready For Sale
14	52408180	Silver Spring	MD	Single Family	4	2.5	1995	Good	495880.00	Ready For Sale
15	68830614	College Park	MD	Single Family	4	3.5	2000	Excellent	620724.00	Ready For Sale
16	81115599	Silver Spring	MD	Condominium	1	1	2000	Good	252775.00	Ready For Sale
17	92417926	Washington	DC	Single Family	5	3.5	2000	Good	555885.00	Ready For Sale
18	92747400	Chevy Chase	MD	Townhouse	3	2.5	1992	Bad Shape	415665.00	Ready For Sale
19	96114604	Laurel	MD	NULL	NULL	NULL	NULL	Good	422625.00	Sold
20	97172037	Alexandria	NULL	Single Family	3	NULL	1965	NULL	345660.00	Sold

5. به منظور تنظیم یک شرط، برای مثال جهت مشاهده ی املاکی که در حال حاضر برای فروش گذاشته شده اند(مقدار فیلد status متناظر آن ها Ready for Sale می باشد) ، دستور را به صورت زیر اصلاح نمایید:

```
USE RealEstate1;
GO
SELECT props.PropertyNumber [Property #],
    props.City, props.[State],
    props.PropertyType [Type],
    props.Bedrooms Beds,
    props.Bathrooms Baths,
    props.YearBuilt [Year],
    props.Condition,
    props.MarketValue "Market Value",
    props.SaleStatus "Status"
FROM Listing.Properties props
WHERE props.SaleStatus = N'Ready For Sale';
GO
```

6. برای اجرای دستورات، کلید F5 را بزنید.

7. همان طور که مشاهده می کنید، 10 رکورد از املاکی که آماده ی فروش هستند به نمایش در می آید:

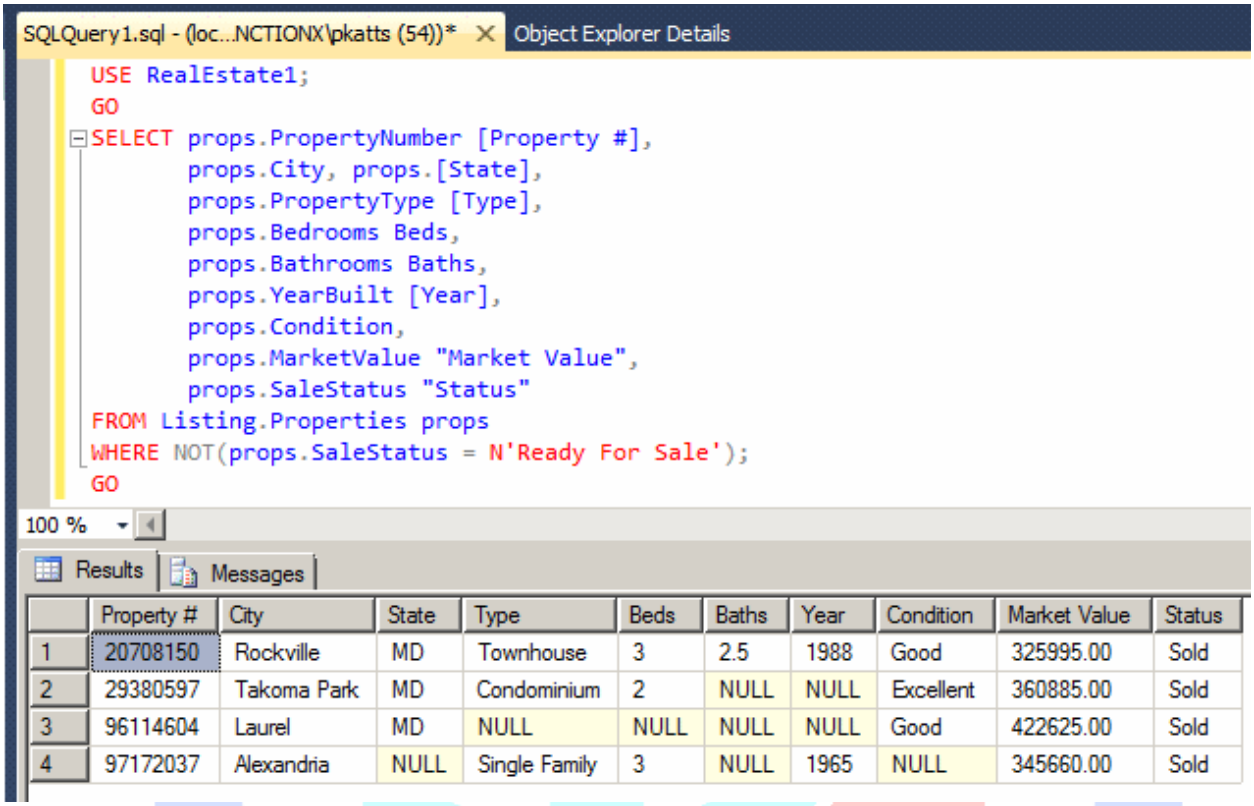


	Property #	City	State	Type	Beds	Baths	Year	Condition	Market Value	Status
1	24747299	Silver Spring	MD	Single Family	3	3	1996	Excellent	625450.00	Ready For Sale
2	29407906	College Park	MD	Single Family	5	2.5	1995	Excellent	485995.00	Ready For Sale
3	29744618	Laurel	MD	Townhouse	4	1.5	2002	Needs Repair	412885.00	Ready For Sale
4	42081159	NULL	DC	NULL	2	NULL	1982	NULL	312555.00	Ready For Sale
5	44759760	Hyattsville	MD	Townhouse	3	2	1992	Excellent	365880.00	Ready For Sale
6	52408180	Silver Spring	MD	Single Family	4	2.5	1995	Good	495880.00	Ready For Sale
7	68830614	College Park	MD	Single Family	4	3.5	2000	Excellent	620724.00	Ready For Sale
8	81115599	Silver Spring	MD	Condominium	1	1	2000	Good	252775.00	Ready For Sale
9	92417926	Washington	DC	Single Family	5	3.5	2000	Good	555885.00	Ready For Sale
10	92747400	Chevy Chase	MD	Townhouse	3	2.5	1992	Bad Shape	415665.00	Ready For Sale

8. حال برای بازیابی املاکی که در حال حاضر آماده ی فروش نیستند (مقدار فیلد متناظر Status آن ها برابر با sold می باشد)، دستور را به صورت زیر اصلاح نمایید:

```
USE RealEstate1;
GO
SELECT props.PropertyNumber [Property #],
    props.City, props.[State],
    props.PropertyType [Type],
    props.Bedrooms Beds,
    props.Bathrooms Baths,
    props.YearBuilt [Year],
    props.Condition,
    props.MarketValue "Market Value",
    props.SaleStatus "Status"
FROM Listing.Properties props
WHERE NOT(props.SaleStatus = N'Ready For Sale');
GO
```

9. به منظور اجرای دستور، کلید F5 را بزنید. در پی اجرای کوئری، سه رکورد واکشی می شود.



	Property #	City	State	Type	Beds	Baths	Year	Condition	Market Value	Status
1	20708150	Rockville	MD	Townhouse	3	2.5	1988	Good	325995.00	Sold
2	29380597	Takoma Park	MD	Condominium	2	NULL	NULL	Excellent	360885.00	Sold
3	96114604	Laurel	MD	NULL	NULL	NULL	NULL	Good	422625.00	Sold
4	97172037	Alexandria	NULL	Single Family	3	NULL	1965	NULL	345660.00	Sold

10. داخل Query Editor کلیک نموده و Ctrl + A را فشار دهید.

همان طور که می دانید از وضعیت املاکی که مقدار متناظر آن ها در ستون Status، null می باشد آگاهی نداریم، بدین معنی که مطمئن نیستیم آیا آن ملک خاص آماده ی فروش هست، به فروش رسیده یا خیر. اکنون می خواهیم تمامی املاکی که در حال حاضر آماده ی فروش نیستند و همچنین آن دسته از املاکی که وضعیت فروش آن ها مشخص نیست را استخراج کرده و به نمایش بگذاریم. برای نیل به این هدف، دستور زیر را تایپ نمایید:

```
CREATE DATABASE RealEstate2;
GO
USE RealEstate2;
GO
CREATE SCHEMA Listing;
GO
CREATE TABLE Listing.PropertiesTypes
(
   PropertyType nvarchar(20) PRIMARY KEY not null,
   [Description] nvarchar(max),
);
GO
CREATE TABLE Listing.PropertiesConditions
(
   Condition nvarchar(20),
   [Description] nvarchar(max),
   CONSTRAINT PK_PropertiesConditions PRIMARY KEY(Condition)
);
GO
CREATE TABLE Listing.SalesStatus
(
```

```
    SaleStatus nvarchar(20),
    [Description] nvarchar(max),
    CONSTRAINT PK_SalesStatus PRIMARY KEY(SaleStatus)
);
GO

INSERT INTO Listing.PropertiesTypes
VALUES(N'Unknown', N'The property type was not specified or was not known'),
    (N'Condominium', N'A condominium, also called condo, is a unit built in a small, medium, or large building. It ressembles an apartment. It may have one, two, or more bedrooms.'),
    (N'Townhouse', N'A townhouse, sometimes called a town house or town home, is a relatively small house attached to at least another house.'),
    (N'Single Family', N'A single family is a stand-alone house. It may have one, two or three levels, also called stories.');
GO
INSERT INTO Listing.PropertiesConditions
VALUES(N'Unknown', N'Unknown property condition'),
    (N'Excellent', N'An excellent property is one that has everything perfect or almost. There are no major repairs to be made.'),
    (N'Good', N'A property is good if it is good enough to be sold. It may be less than perfect but it is wholly acceptable.'),
    (N'Needs Repair', N'This type of condition indicates that one or more repairs are necessairy. The property in this condition is not ready for sale.'),
    (N'Bad Shape', N'A property is in bad shape if it requires a mojor or many repairs.');
GO
INSERT Listing.SalesStatus
VALUES(N'N/A', N'The sale status of this property is not definitely spaecified'),
    (N'Ready For Sale', N'The property is currently available for sale.'),
    (N'Sold', N'The property has been sold.');
GO
CREATE TABLE Listing.Properties
(
    PropertyNumber nvarchar(12),
    [Address] nvarchar(60),
    City nvarchar(50),
    [State] nchar(2),
    ZIPCode nvarchar(12),
    PropertyType nvarchar(20) default N'Unknown',
    Bedrooms smallint,
    Bathrooms float,
    Stories smallint,
    FinishedBasement bit,
    IndoorGarage tinyint,
    YearBuilt smallint,
    Condition nvarchar(20) default N'Unknown',
    MarketValue money,
    SaleStatus nvarchar(20) default N'N/A',
    CONSTRAINT PK_Properties PRIMARY KEY(PropertyNumber),
        CONSTRAINT FK_PropertiesTypes FOREIGN KEY(PropertyType) REFERENCES
Listing.PropertiesTypes(PropertyType),
        CONSTRAINT FK_PropertiesConditions FOREIGN KEY(Condition) REFERENCES
Listing.PropertiesConditions(Condition),
    CONSTRAINT FK_SalesStatus FOREIGN KEY(SaleStatus) REFERENCES Listing.SalesStatus(SaleStatus)
);
GO

INSERT INTO Listing.Properties
VALUES(N'52408180', N'1640 Lombardo Ave', N'Silver Spring', N'MD', N'20904', N'Single Family', 4, 2.5, 3, 1, 2, 1995,
N'Good', 495880.00, N'Ready For Sale'),
    (N'68830614', N'10315 North Hacht Rd', N'College Park', N'MD', N'20747', N'Single Family', 4, 3.5, 3, 1, 2, 2000,
N'Excellent', 620724.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State],
  ZIPCode, Condition, FinishedBasement, Stories, MarketValue, SaleStatus)
VALUES(N'96114604', N'6366 Lolita Drive', N'Laurel', N'MD', N'20707', N'Good', 1, 2, 422625.00, N'Sold');
```

```
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, PropertyType, Bedrooms, IndoorGarage, MarketValue)
VALUES(N'39485797', N'9002 Palasko Hwy', N'Tysons Corner', N'Condominium', 2, 2, 422895.00);
GO
INSERT INTO Listing.Properties(PropertyNumber, [State], ZIPCode, Bedrooms, YearBuilt, MarketValue, SaleStatus)
VALUES(N'42081159', N'DC', N'20011', 2, 1982, 312555, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, ZIPCode, PropertyType, Bedrooms, YearBuilt, MarketValue,
SaleStatus)
VALUES(N'97172037', N'Alexandria', N'22024', N'Single Family', 3, 1965, 345660.00, N'Sold');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State], PropertyType, Condition, Bedrooms, Bathrooms,
MarketValue)
VALUES(N'20880417', N'4140 Holisto Crt', N'Germantown', N'MD', N'Condominium', N'Excellent', 2, 1, 215495.00);
GO
INSERT INTO Listing.Properties
VALUES(N'92747400', N'9522 Lockwood Rd', N'Chevy Chase', N'MD', N'20852', N'Townhouse', 3, 2.5, 3, 0, 1,  1992, N'Bad
Shape', 415665.00, N'Ready For Sale'),
    (N'20708150', N'14250 Parkdoll Rd', N'Rockville', N'MD', N'20854', N'Townhouse', 3, 2.5, 2, 1, 0, 1988, N'Good',
325995.00, N'Sold');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, PropertyType, Bedrooms, YearBuilt, MarketValue)
VALUES(N'29304857', N'Washington', N'Townhouse', 4, 1975, 366775.00);
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State],
        ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms, YearBuilt, MarketValue)
VALUES(N'28085040', N'10340 Helmes Street #408', N'Silver Spring', N'MD', N'20906', N'Condominium', N'Good', 1, 1, 2000,
242775.00);
GO
INSERT INTO Listing.Properties
VALUES(N'24747299', N'1008 Coppen Street', N'Silver Spring', N'MD', N'20906', N'Single Family', 3, 3, 3, 1, 3, 1996,
N'Excellent', 625450.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, ZIPCode, PropertyType, Stories, YearBuilt, MarketValue)
VALUES(N'39485070', N'Chevy Chase', N'20956', N'Single Family', 3, 2001, 525450.00);
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State], PropertyType, Condition, Bedrooms, MarketValue,
SaleStatus)
VALUES(N'29380597', N'686 Herod Ave #D04', N'Takoma Park', N'MD', N'Condominium', N'Excellent', 2, 360885.00,
N'Sold');
GO
INSERT INTO Listing.Properties
VALUES(N'29744618', N'14005 Sniders Blvd', N'Laurel', N'MD', N'20707', N'Townhouse', 4, 1.5, 3, 1, 0, 2002, N'Needs
Repair', 412885.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, City, ZIPCode, Condition, Bedrooms, Stories, YearBuilt)
VALUES(N'28074085', N'Silver Spring', N'20905', N'Good', 4, 2, 1965);
GO
INSERT INTO Listing.Properties
VALUES(N'92417926', N'680 Prushia Rd NE', N'Washington', N'DC', N'20008', N'Single Family', 5, 3.5, 3, 0, 3, 2000, N'Good',
555885.00, N'Ready For Sale'),
    (N'29407906', N'14688 Parrison Street', N'College Park', N'MD', N'20742', N'Single Family', 5, 2.5, 2, 1, 2, 1995,
N'Excellent', 485995.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties(PropertyNumber, [Address], City, [State],
  ZIPCode, PropertyType, Condition, Bedrooms, Bathrooms, YearBuilt, MarketValue, SaleStatus)
VALUES(N'81115599', N'10340 Helmes Street #1012', N'Silver Spring',
    N'MD', N'20906', N'Condominium', N'Good', 1, 1, 2000, 252775.00, N'Ready For Sale');
GO
INSERT INTO Listing.Properties
VALUES(N'44759760', N'4201 Vilamar Ave', N'Hyattsville', N'MD', N'20782', N'Townhouse', 3, 2, 2, 1, 2, 1992, N'Excellent',
365880.00, N'Ready For Sale');
```

```
GO
```

11. داخل Query Editor کلیک کرده و Ctrl + A را بزنید.

12. برای نمایش لیست کامل املاک، کد زیر را درج و اجرا نمایید:

```
USE RealEstate2;
GO
SELECT PropertyNumber [Property #], City, [State],
    PropertyType [Type], Bedrooms Beds,
    Bathrooms Baths, YearBuilt [Year],
    Condition, MarketValue "Market Value", SaleStatus "Status"
FROM Listing.Properties;
GO
```

13. همان طور که مشاهده می کنید دیگر فیلدی حاوی مقدار NULL در ستون Type، Condition، Status جدول جاری مشاهده نمی شود:

SQLQuery1.sql - (loc...NCTIONX\pkatts (55))* | Object Explorer Details

```
USE RealEstate2;
GO
SELECT props.PropertyNumber [Property #],
       props.City, props.[State],
       props.PropertyType [Type],
       props.Bedrooms Beds,
       props.Bathrooms Baths,
       props.YearBuilt [Year],
       props.Condition,
       props.MarketValue "Market Value",
       props.SaleStatus "Status"
FROM Listing.Properties props;
GO
```

100 %

Results | Messages

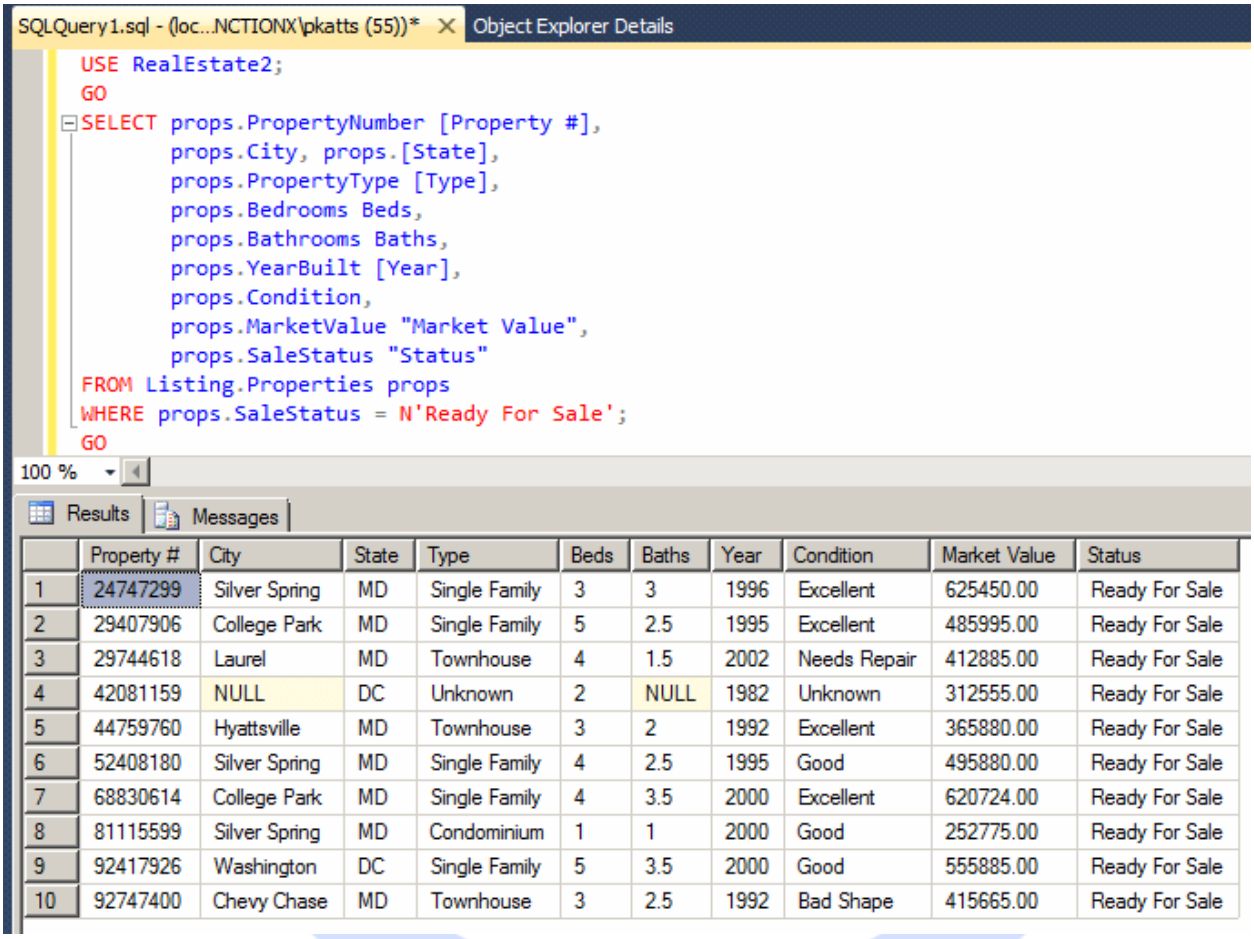
	Property #	City	State	Type	Beds	Baths	Year	Condition	Market Value	Status
1	20708150	Rockville	MD	Townhouse	3	2.5	1988	Good	325995.00	Sold
2	20880417	Germantown	MD	Condominium	2	1	NULL	Excellent	215495.00	N/A
3	24747299	Silver Spring	MD	Single Family	3	3	1996	Excellent	625450.00	Ready For Sale
4	28074085	Silver Spring	NULL	Unknown	4	NULL	1965	Good	NULL	N/A
5	28085040	Silver Spring	MD	Condominium	1	1	2000	Good	242775.00	N/A
6	29304857	Washington	NULL	Townhouse	4	NULL	1975	Unknown	366775.00	N/A
7	29380597	Takoma Park	MD	Condominium	2	NULL	NULL	Excellent	360885.00	Sold
8	29407906	College Park	MD	Single Family	5	2.5	1995	Excellent	485995.00	Ready For Sale
9	29744618	Laurel	MD	Townhouse	4	1.5	2002	Needs Repair	412885.00	Ready For Sale
10	39485070	Chevy Chase	NULL	Single Family	NULL	NULL	2001	Unknown	525450.00	N/A
11	39485797	Tysons Corner	NULL	Condominium	2	NULL	NULL	Unknown	422895.00	N/A
12	42081159	NULL	DC	Unknown	2	NULL	1982	Unknown	312555.00	Ready For Sale
13	44759760	Hyattsville	MD	Townhouse	3	2	1992	Excellent	365880.00	Ready For Sale
14	52408180	Silver Spring	MD	Single Family	4	2.5	1995	Good	495880.00	Ready For Sale
15	68830614	College Park	MD	Single Family	4	3.5	2000	Excellent	620724.00	Ready For Sale
16	81115599	Silver Spring	MD	Condominium	1	1	2000	Good	252775.00	Ready For Sale
17	92417926	Washington	DC	Single Family	5	3.5	2000	Good	555885.00	Ready For Sale
18	92747400	Chevy Chase	MD	Townhouse	3	2.5	1992	Bad Shape	415665.00	Ready For Sale
19	96114604	Laurel	MD	Unknown	NULL	NULL	NULL	Good	422625.00	Sold
20	97172037	Alexandria	NULL	Single Family	3	NULL	1965	Unknown	345660.00	Sold

14. اکنون برای مشاهده ی آن املاکی که آماده ی فروش هستند (که مقدار متناظر فیلدهای Status آن برابر با Ready For Sale می باشد) دستور را به ترتیب زیر اصلاح نمایید:

```
USE RealEstate2;
GO
SELECT props.PropertyNumber [Property #],
    props.City, props.[State],
    props.PropertyType [Type],
    props.Bedrooms Beds,
    props.Bathrooms Baths,
    props.YearBuilt [Year],
    props.Condition,
    props.MarketValue "Market Value",
    props.SaleStatus "Status"
FROM Listing.Properties props
```

```
WHERE props.SaleStatus = N'Ready For Sale';
GO
```

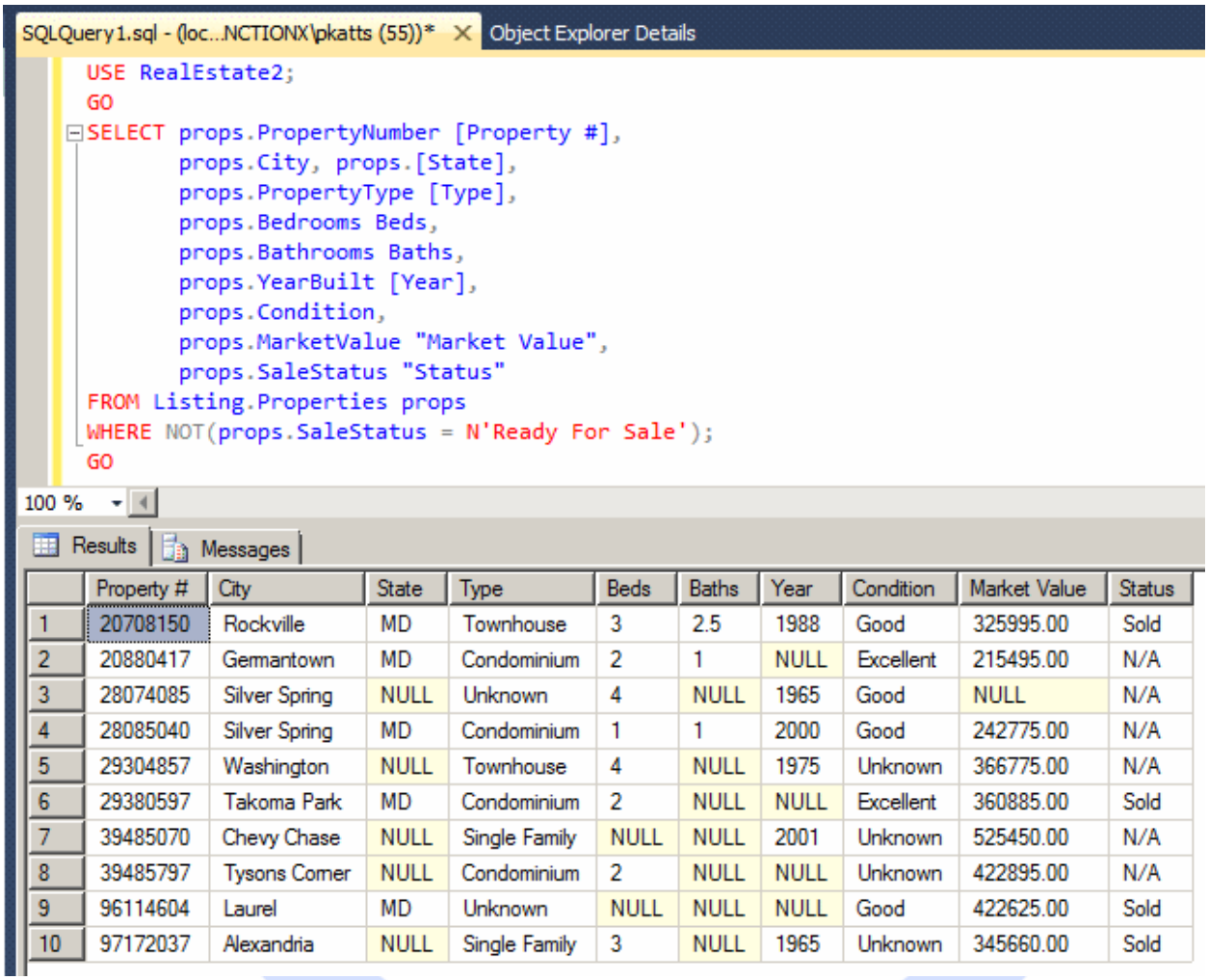
15. دستور را اجرا نمایید. همان طور که می بینید، 10 رکورد از جدول به نمایش در می آیند که مقادیر ستون status آن ready for sale می باشد:



	Property #	City	State	Type	Beds	Baths	Year	Condition	Market Value	Status
1	24747299	Silver Spring	MD	Single Family	3	3	1996	Excellent	625450.00	Ready For Sale
2	29407906	College Park	MD	Single Family	5	2.5	1995	Excellent	485995.00	Ready For Sale
3	29744618	Laurel	MD	Townhouse	4	1.5	2002	Needs Repair	412885.00	Ready For Sale
4	42081159	NULL	DC	Unknown	2	NULL	1982	Unknown	312555.00	Ready For Sale
5	44759760	Hyattsville	MD	Townhouse	3	2	1992	Excellent	365880.00	Ready For Sale
6	52408180	Silver Spring	MD	Single Family	4	2.5	1995	Good	495880.00	Ready For Sale
7	68830614	College Park	MD	Single Family	4	3.5	2000	Excellent	620724.00	Ready For Sale
8	81115599	Silver Spring	MD	Condominium	1	1	2000	Good	252775.00	Ready For Sale
9	92417926	Washington	DC	Single Family	5	3.5	2000	Good	555885.00	Ready For Sale
10	92747400	Chevy Chase	MD	Townhouse	3	2.5	1992	Bad Shape	415665.00	Ready For Sale

16. حال برای مشاهده ی (رکوردهای) املاکی که آماده ی فروش نیستند (مقدار فیلدهای ستون status آن ها sold یا N/A می باشد)، کد را به شکل زیر تغییر دهید:

```
USE RealEstate2;
GO
SELECT props.PropertyNumber [Property #],
    props.City, props.[State],
    props.PropertyType [Type],
    props.Bedrooms Beds,
    props.Bathrooms Baths,
    props.YearBuilt [Year],
    props.Condition,
    props.MarketValue "Market Value",
    props.SaleStatus "Status"
FROM Listing.Properties props
WHERE NOT(props.SaleStatus = N'Ready For Sale');
GO
```



	Property #	City	State	Type	Beds	Baths	Year	Condition	Market Value	Status
1	20708150	Rockville	MD	Townhouse	3	2.5	1988	Good	325995.00	Sold
2	20880417	Germantown	MD	Condominium	2	1	NULL	Excellent	215495.00	N/A
3	28074085	Silver Spring	NULL	Unknown	4	NULL	1965	Good	NULL	N/A
4	28085040	Silver Spring	MD	Condominium	1	1	2000	Good	242775.00	N/A
5	29304857	Washington	NULL	Townhouse	4	NULL	1975	Unknown	366775.00	N/A
6	29380597	Takoma Park	MD	Condominium	2	NULL	NULL	Excellent	360885.00	Sold
7	39485070	Chevy Chase	NULL	Single Family	NULL	NULL	2001	Unknown	525450.00	N/A
8	39485797	Tysons Corner	NULL	Condominium	2	NULL	NULL	Unknown	422895.00	N/A
9	96114604	Laurel	MD	Unknown	NULL	NULL	NULL	Good	422625.00	Sold
10	97172037	Alexandria	NULL	Single Family	3	NULL	1965	Unknown	345660.00	Sold

17. برای اجرای کد، کلید F5 را فشار دهید. این بار تمامی املاکی که آماده ی فروش نیستند نمایش داده می شوند.

نتیجه می گیریم که در ستون کلید خارجی نباید فیلد تهی یا مقدار NULL وجود داشته باشد.

پشتیبانی و قیود

مسائل متعددی وجود دارد که باید در کار با یک پایگاه داده ی رابطه ای به آن توجه کرد. برخی از این مسائل به ستون های جدول و برخی دیگر به رکوردها مربوط می شوند. بنابراین بایستی در زمان افزودن ستون های جدید یا حذف رکوردهای جدول دقت ویژه به ساختار آن جدول داشت.

اگر تصمیم به حذف جدول دارید، ابتدا می بایست بررسی کنید آن جدول با جدول دیگری ارتباط دارد یا خیر. همچنین آیا جدول مورد نظر یک جدول پدر (رکوردهای خود را در اختیار جدول دیگر قرار می دهد) است یا فرزند (یکی از ستون

های آن مقادیرش را از جدول پدر می گیرد)؟ اگر جدول فرزندی داشته باشد، در آن صورت می توان به راحتی آن را حذف کرد. اما چنانچه جدول مورد نظر یک جدول پدر باشد، در آن صورت خطا صادر می شود.

اعمال محدودیت کلید اصلی (Primary Key)

پس از ایجاد جدول یا در زمان ارث بری یک جدول که توسط کاربر دیگری ایجاد شده، ناگهان متوجه می شوید که کلید اصلی برای آن جدول تعریف نشده است. برای حل این مشکل کافی است با پیروی از اصول خاصی یک ستون کلید اصلی به جدول پدر اضافه نمایید. برای این منظور، دو راه پیشرو دارید:

تصور کنید جدول زیر را ایجاد کرده اید:

```
CREATE TABLE Employees
(
   FirstName nvarchar(20),
   LastName nvarchar(20),
   DepartmentCode nchar(6)
);
GO
```

می توانید عبارت PRIMARY KEY را بلافاصله پس از تعریف ستون جدید اضافه نمایید (کلیدواژه ی ADD، اسم ستون، نوع داده ای آن، عبارت not null و در نهایت PRIMARY KEY را درج نمایید). مثال:

```
ALTER TABLE Employees
ADD EmployeeNumber int not null PRIMARY KEY;
GO
```

همچنین می توانید یک ستون به جدول اضافه کرده و سپس فرمول CONSTRAINT، اسم کلید اصلی، عبارت کلیدی PRIMARY KEY و داخل پرانتز اسم ستون را ذکر نمایید.

```
ALTER TABLE Employees
ADD EmployeeNumber int not null
CONSTRAINT PK_Employees PRIMARY KEY(EmployeeNumber);
GO
```

اعمال قید کلید خارجی (Foreign Key)

همان طور که یک کلید اصلی به جدول جاری اضافه کردیم، حال یک ستون جدید که کلید خارجی محسوب می شود اضافه می کنیم. در مثال زیر یک جدول فرزند به نام Persons می سازیم:

```
CREATE TABLE Genders
(
   GenderID int not null PRIMARY KEY,
   Gender nvarchar(20)
);
GO
CREATE TABLE Persons
```

```
(
   PersonID int PRIMARY KEY NOT NULL,
   FirstName nvarchar(20),
   LastName nvarchar(20) NOT NULL
);
GO
```

فرمول افزودن یک قید کلید خارجی به جدول جاری به صورت زیر می باشد:

```
ALTER TABLE TableName
ADD NewColumnName DataType Options
   FOREIGN KEY REFERENCES ParentTableName(ColumnNameOfOtherTable);
GO
```

کد زیر یک کلید خارجی به جدول Persons اضافه می کند:

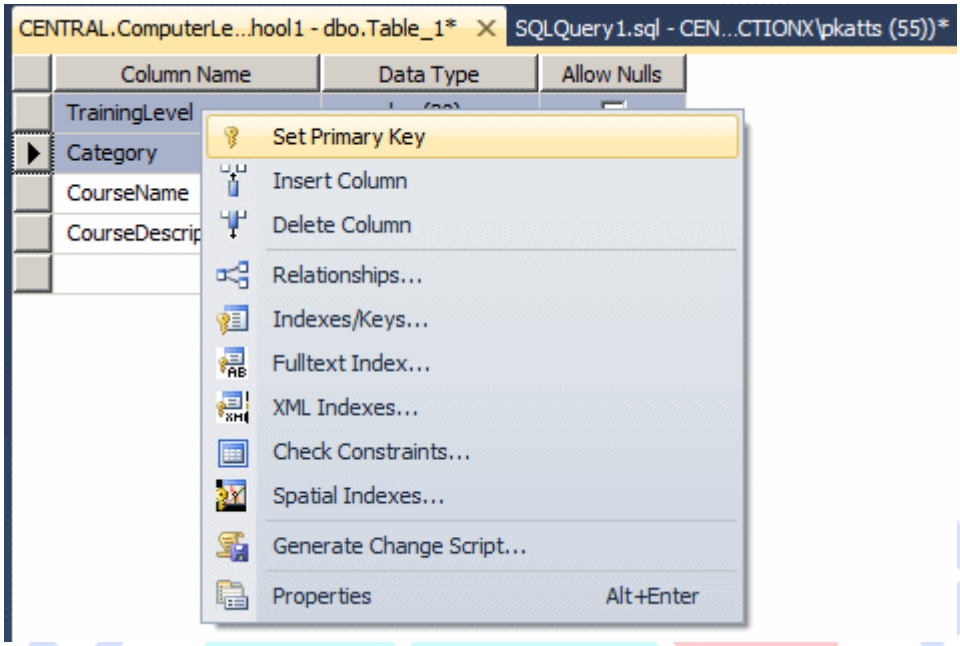
```
ALTER TABLE Persons
ADD GenderID int NULL FOREIGN KEY REFERENCES Genders(GenderID);
GO
```

کلیدهای مرکب (Composite Keys)

تاکنون تنها یک ستون از جدول را به عنوان کلید اصلی درنظر می گرفتیم. گاهی یک ستون به تنهایی نمی تواند یک رکورد را منحصر بفرد کند. برای این منظور بیش از یک ستون را کلید اصلی قرار می دهیم. کلید مرکب در واقع یک نوع کلید اصلی است که بیش از یک ستون را دربرمی گیرد.

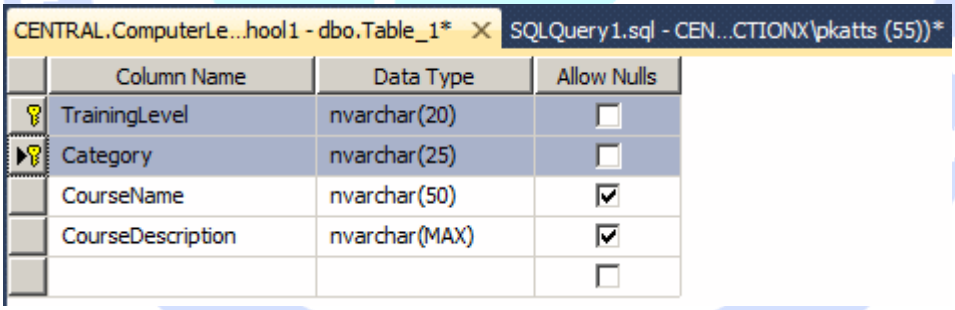
برای ایجاد یک کلید مرکب به صورت ویزاردی، در زمان ساخت و طراحی (design) جدول، کلید Ctrl را گرفته و سپس بر روی row header (کادر واقع در سمت چپ فیلد) ستون های مورد نظر (آنهایی که قرار است کلید اصلی شوند) کلیک نمایید (تا هر دو سطر آبی شوند). کلید Ctrl را رها کرده، سپس:

1. بر روی ستون های انتخابی راست کلیک کرده و گزینه ی Set Primary Key را انتخاب نمایید.



2. و یا: در نوار ابزار Table Designer، بر روی دکمه ی  کلیک کنید.

در هر دو مورد، آیکون کلید  بر روی row header ستون های کلید اصلی، پدیدار می شود.



به همین نحو، می توان تعداد بی شماری ستون را به عنوان کلید اصلی یک جدول انتخاب کرد. یکی از روش های ایجاد کلید اصلی نیز استفاده از کلید خارجی است که کلید اصلی در جدول دیگر هستند.

برای ایجاد یک کلید اصلی مرکب با کدنویسی، داخل پرانتز عبارت PRIMARY KEY، اسم ستون ها را ذکر کرده و توسط ویرگول از هم جدا نمایید. مثال:

```
CREATE DATABASE InformationTechnologyJournal;
GO
USE InformationTechnologyJournal;
GO

CREATE SCHEMA Publishing;
GO
CREATE SCHEMA Authorship;
GO
```

```
CREATE TABLE Authorship.Reviewers
(
	ReviewerNumber nchar(6) not null,
	FirstName nvarchar(24),
	MiddleName nvarchar(24),
	LastName nvarchar(24),
	Citizenship nvarchar(40),
	Constraint PK_Reviewers Primary Key(ReviewerNumber)
);
GO
CREATE TABLE Publishing.Affiliations
(
	AffiliationCode nchar(5) not null,
	AffiliationName nvarchar(60),
	Constraint PK_Affiliations Primary Key(AffiliationCode)
);
GO
CREATE TABLE Authorship.ReviewersAffiliations
(
	ReviewerNumber nchar(6) not null,
	AffiliationCode nchar(5) not null,
	Constraint PK_ReviewersAffiliations Primary Key(ReviewerNumber, AffiliationCode)
);
GO
```

Candidate Key (ستون کلید اصلی)

کلید کاندید به ستونی اطلاق می شود که کلید اصلی انتخاب شده است. اگر کلید اصلی یک ستون را شامل شود، آن ستون کلید کاندید تلقی می شود. در صورتی که چند ستون کلید اصلی انتخاب شوند، هر یک از ستون ها یک کلید کاندید محسوب می شود.