



عنوان مقاله: دستور Create Table در SQL Server - آشنایی با کاربردها و نحوه ایجاد جدول

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۶ اسفند ۱۴۰۲

منبع: <https://nikamooz.com/create-table-command-in-sql-server/>

دستور Create Table در SQL Server یکی از دستورهای کاربردی در این سیستم مدیریت پایگاه داده (RDBMS) به حساب می‌آید. یادگیری نحوه استفاده از این دستور، به شما مهارت‌های ارزشمندی در زمینه سازماندهی و مدیریت داده‌ها می‌بخشد و به‌واسطه آن می‌توانید به ایجاد جدول در SQL Server بپردازید. در مقالات قبلی به بررسی کامل **دستور Select** و **دستور Insert** و **دستور Update**، **عبارت Where** و **نحوه تعریف محدودیت (Constraint)** و چگونگی به کار بردن هر یک پرداختیم. در این مطلب، می‌خواهیم به آموزش نحوه ایجاد جدول از طریق دستور Create Table در SQL Server بپردازیم تا شما آن را از طریق کوئری‌ها و مثال‌های قابل درک بیاموزید.

### دستور SQL Server چیست و چه کاربردی دارد؟

دستورات SQL Server عباراتی هستند که به شما امکان تعامل با پایگاه داده Microsoft SQL Server را می‌دهند. به‌واسطه این دستورات، اقداماتی همچون ایجاد و مدیریت پایگاه داده، تجزیه و تحلیل دیتا، مدیریت دسترسی کاربران، استخراج و دستکاری داده‌ها امکان‌پذیر خواهد بود. برای آشنایی بیشتر با مباحث پراهمیت SQL Server، کاربردها، **انواع RDBMS**، مشاغل مرتبط با این حوزه و سایر موارد، پیشنهاد می‌شود **مقاله جامع آموزش SQL Server** را مطالعه کنید.



## معرفی دستور Create Table در SQL Server

دستور Create Table در SQL Server ، به منظور تعریف و ساخت جدول جدید در پایگاه داده به کار می‌رود. ساختار کلی دستور Create Table به صورت زیر است:

```
CREATE TABLE [database_name.][schema_name.]table_name (
    column_name1 data_type [NOT NULL],
    column_name2 data_type,
    ...
);
```

بخش‌های مختلف ساختار فوق، به شرح زیر است:

- **database\_name**: این بخش از دستور Create Table در SQL Server ، انتخابی است؛ یعنی با تعیین آن می‌توانید نام دیتابیس را مشخص کنید که می‌خواهید جدول در آن ساخته شود.
- **schema\_name**: این مورد نیز انتخابی است و به واسطه آن، Schema ای تعریف می‌شود که در آن‌جا، جدول پایگاه داده ساخته خواهد شد. اگر schema\_name را در زمان ساخت جدول در SQL Server تعیین نکنید، به طور پیش‌فرض جدول جدید در Schema کاربر ساخته می‌شود و فقط توسط آن User خاص قابل مشاهده خواهد بود.
- **table\_name**: این جز از دستور Create Table الزامی است و بیان‌گر نامی است که به جدول جدید اختصاص می‌دهید.
- **column\_definition**: این جز هر ستون از جدول، شامل نام، نوع داده و Nullity آن را تعریف می‌کند. منظور از Nullity، پوچ یا نبودن مقدار ستون است.

## ملاحظات جانبی

بهتر است به ملاحظات زیر در خصوص دستور Create Table در SQL Server توجه کنید:

- شما می‌توانید با اعمال محدودیت (Constraint) روی ستون‌ها یا تمام جدول، آن‌ها را محدودتر کنید.
- امکان ایجاد جداول Temporal از طریق # یا ## در ابتدای نام جدول وجود دارد.
- امکانات پیشرفته متعددی در زمان ساخت جدول در SQL Server وجود دارد؛ مواردی همچون تعیین Filegroup ها و جداول Memory-Optimized از این قابلیت‌ها به شمار می‌روند.

## کاربرد دستور Create Table در SQL Server

دستور Create Table در SQL Server کاربردها و موارد استفاده مشخصی دارد که در ادامه بررسی می‌شوند.

### ۱- ذخیره سازی داده های جدید

- **ساخت پایگاه داده جدید:** هنگام ایجاد یک دیتابیس جدید، شما از دستور Create Table برای تعریف جداول نگهدارنده داده‌های خود استفاده خواهید کرد. این فرآیند، تعیین مواردی مانند ستون‌ها، **انواع نوع داده**، محدودیت‌ها و **انواع کلید** را شامل می‌شود.
- **درج جدول به پایگاه داده موجود:** با تکامل ضرورت‌های مربوط به داده‌ها، این امکان وجود دارد که جداول جدیدی به هدف نگهداری اطلاعات جدید ایجاد شوند. در چنین شرایطی، ممکن است به ذخیره‌سازی اطلاعاتی همچون جزئیات اضافی مشتریان، رکوردهای تراکنشی و اطلاعات محصول نیاز باشد.

### ۲- Import داده از سایر منابع

- **بارگذاری داده‌ها از فایل:** شما می‌توانید با کمک دستور Create Table در SQL Server به همراه یک عبارت Select، داده‌ها را از فایل‌های اکسل، CSV یا سایر فرمت‌های فایلی Import کنید. چنین امکانی به شما اجازه می‌دهد جداول خود را به همراه داده‌ها آن، به صورت کارآمد در دیتابیس قرار دهید.
- **مایگریت داده‌ها از سایر دیتابیس‌ها:** در شرایطی که شما بخواهید داده‌ها از **انواع پایگاه داده** دیگر به یک سیستم دیتابیس جدید مهاجرت پیدا کنند، باید جداول را با ساختارهای سازگار با دیتابیس SQL Server ایجاد کنید.

### ۳- مدل سازی و ساختاردهی مجدد داده ها

- **بهینه‌سازی طراحی جداول:** شما می‌توانید با ساخت جدول در SQL Server و بهبود ساختار آن‌ها، کارایی دیتابیس را بهینه‌سازی کنید. این فرآیند، احتمالاً اقداماتی مانند جداسازی جداول فعلی پایگاه داده، ساخت روابط (Relationships) جدید و **نرمال سازی داده ها** (Data Normalization) را دربرمی‌گیرد.
- **ساخت جدول موقت برای تحلیل:** در مواقعی که بخواهید به تجزیه و تحلیل داده‌ها یا تبدیل‌های پیچیده بپردازید، می‌توانید برای ساخت Temporal Table از دستور Create Table در SQL Server بهره‌مند شوید. با کمک این روش، امکان نگهداری نتایج میانی در این جدول موقت فراهم می‌شود و دیگر اثرات حاصل از آن روی جداول اصلی اعمال نخواهند شد.

## ۴- کنترل نسخه و تست نویسی

- **نگهداری نسخه‌های تاریخی از جداول:** برای برخی داده‌های خاص، ممکن است نیاز به نسخه‌های تاریخی وجود داشته باشد. در این شرایط، دستور Create Table در SQL Server به شما این قابلیت را می‌دهد که براساس جداول فعلی، جداول‌های جدید دیگری را ایجاد کنید و از آن‌ها به منظور نگهداری یک وضعیت خاص از دیتا بهره‌مند شوید.
- **ساخت محیط تست:** یکی از کاربردهای دستور Create Table در SQL Server این است که شما می‌توانید یک نسخه کپی از جداول فعلی ایجاد کنید و آن را در یک اسکیمای مستقل و برای اهداف تست و آزمایش مورد استفاده قرار دهید.

## ۵- ایجاد جدول‌های خاص

- **ساخت جدول Memory-Optimized:** برای سناریوهای با کارایی بالا، امکان ساخت **جدول Memory-Optimized** با استفاده از دستور Create Table وجود دارد. این نوع از جداول در حافظه باقی می‌مانند و پردازش و دسترسی به داده‌ها را سرعت می‌بخشند.
  - **ایجاد جدول system-versioned temporal:** این گونه از جداول، به‌طور خاص برای پیگیری تغییرات تاریخی مربوط به داده‌ها به کار می‌روند و اجازه می‌دهند که کاربر داده‌ها را در هر زمان تجزیه و تحلیل کند.
- تا این بخش، به بررسی کاربردهای دستور Create Table در SQL Server پرداخته شد. در ادامه، قصد داریم بررسی کنیم که به چه صورت Create Table به شما امکان ایجاد جدول در SQL Server را می‌دهد.



## نحوه ایجاد جدول در SQL Server

در این بخش، مثال‌هایی از دستور Create Table ارائه خواهیم کرد. برای درک نحوه ایجاد جدول در SQL Server از طریق این دستور، به کوئری‌های زیر توجه کنید.

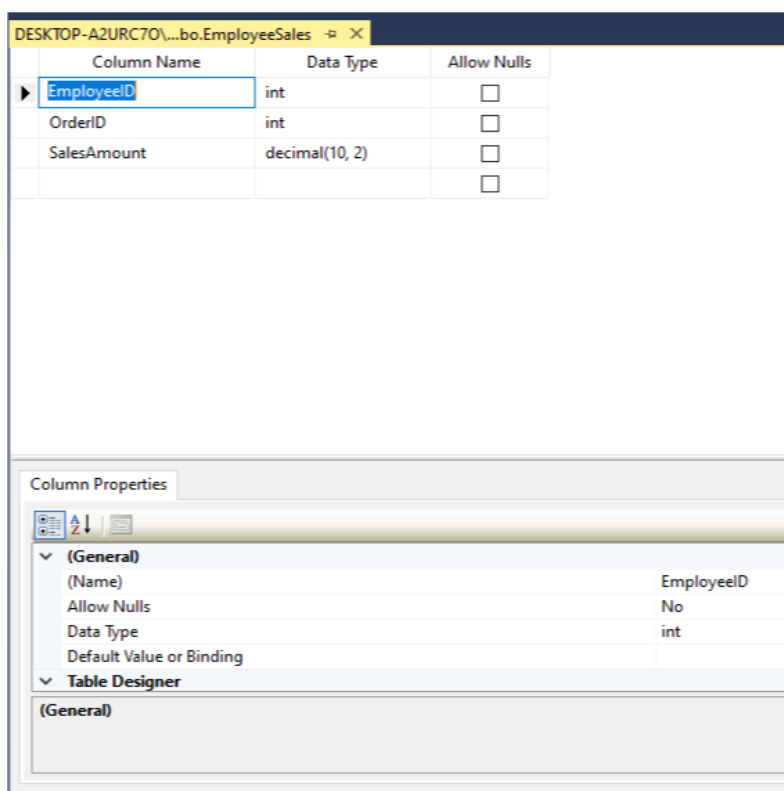
### ایجاد یک جدول ساده

پیش از بررسی بیشتر نحوه ایجاد جدول در SQL Server، توجه کنید که کوئری‌های این بخش همگی، از طریق [دیتابیس Northwind](#) قابل اجرا هستند.

فرض کنید می‌خواهیم یک جدول تحت عنوان EmployeeSales در پایگاه داده Northwind ایجاد کنیم. برای این کار، کفایت کوئری زیر را اجرا کنید:

```
CREATE TABLE EmployeeSales (
    EmployeeID int NOT NULL FOREIGN KEY REFERENCES Employees(EmployeeID),
    OrderID int NOT NULL FOREIGN KEY REFERENCES Orders(OrderID),
    SalesAmount decimal(10,2) NOT NULL
);
```

کوئری فوق، با لینک کردن Employee ID و Order Id به همراه مقدار فروش، اطلاعاتی را درخصوص عملکرد کارمندان فروش ذخیره‌سازی می‌کند. در این جدول، EmployeeID به‌عنوان [کلید خارجی](#) (FK)، به جدول Employees ارجاع داده شده و OrderID نیز یک FK دیگر است که به جدول Orders رفرنس شده است. ضمن اینکه در SalesAmount مقدار فروش هر کارمند ذخیره شده است.



## ایجاد یک جدول با Computed Column

یکی دیگر از کاربردهای دستور Create Table در SQL Server، ایجاد جدولی به همراه Computed Column است. Computed Column یک ستون مجازی محسوب می‌شود که به طور فیزیکی داده‌های خود را ذخیره‌سازی نخواهد کرد؛ در واقع، مقادیر Computed Column، براساس محاسبات، توابع یا محدودیت‌ها، محاسبه و ذخیره‌سازی می‌شوند.

در مثال زیر، قصد داریم یک جدول به نام OrderSummary در پایگاه داده ایجاد کنیم؛ این جدول اطلاعات سفارش را به همراه یک Computed Column برای مبلغ کل سفارش‌ها شامل می‌شود. با توجه به اینکه امکان استفاده از Subquery ها در ستون‌های Computed وجود ندارد، ابتدا یک [تابع تعریف شده توسط کاربر](#) (User-Defined Function) برای محاسبه کل مبلغ یک سفارش ایجاد کرده و سپس آن را برای Computed Column به کار خواهیم برد. برای درک این روند، به اسکریپت زیر توجه کنید:

```
USE Northwind;
GO
```

```
-- Drop the existing function if it exists
IF EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'dbo.CalculateTotalAmount') AND type in (N'FN', N'IF', N'TF',
N'FS', N'FT'))
DROP FUNCTION dbo.CalculateTotalAmount;
GO
```

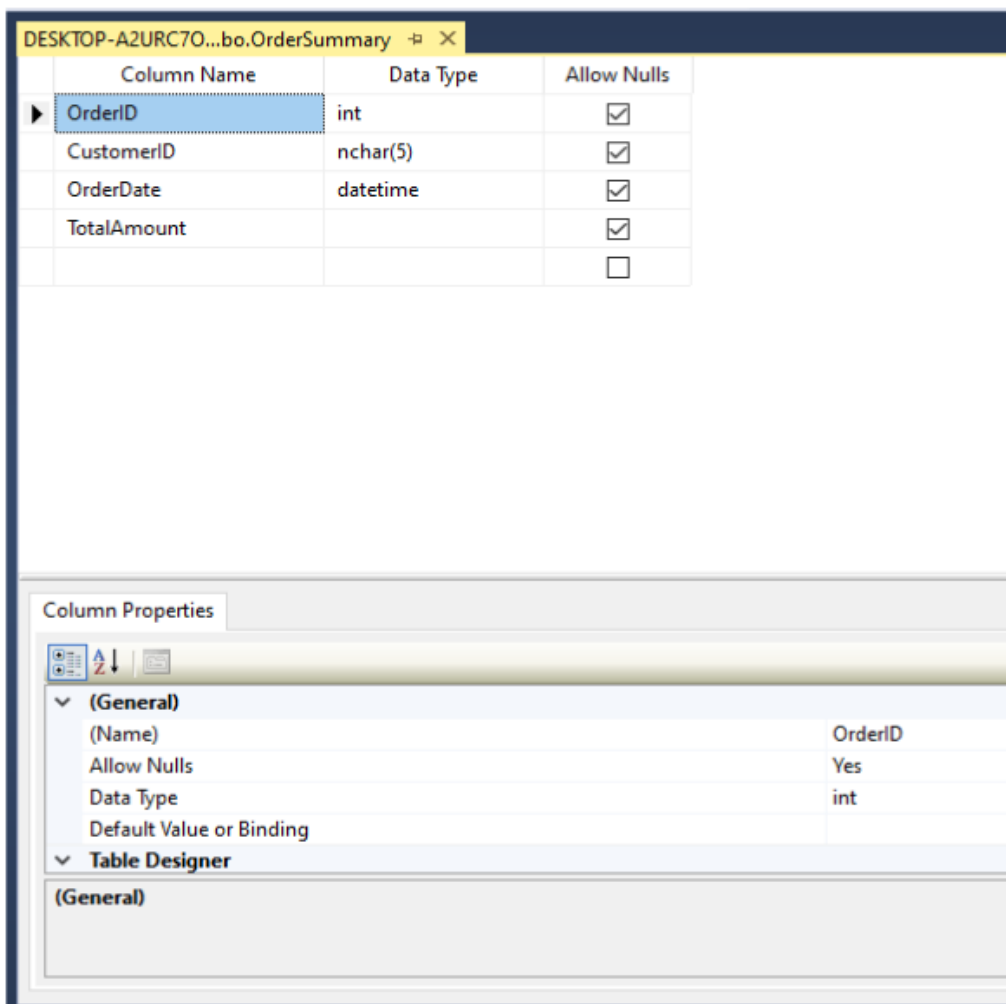
```
-- Create the function to calculate total amount for an order
CREATE FUNCTION dbo.CalculateTotalAmount (@OrderID int)
RETURNS money
AS
BEGIN
    DECLARE @TotalAmount money;

    SELECT @TotalAmount = ISNULL(SUM(UnitPrice * Quantity * (1 -
Discount)), 0)
    FROM [Order Details]
    WHERE OrderID = @OrderID;

    RETURN @TotalAmount;
END;
GO
```

```
-- Create the table with computed column using the function
CREATE TABLE OrderSummary (
    OrderID int,
    CustomerID nchar(5),
    OrderDate datetime,
    TotalAmount AS dbo.CalculateTotalAmount(OrderID)
);
```

پس از اجرای اسکریپت فوق، می‌توانید با مراجعه به فولدر Tables از پایگاه داده Northwind، جدول ایجاد شده را به‌همراه دیزاین آن مشاهده کنید.



### ساخت جدول پیچیده به همراه Join

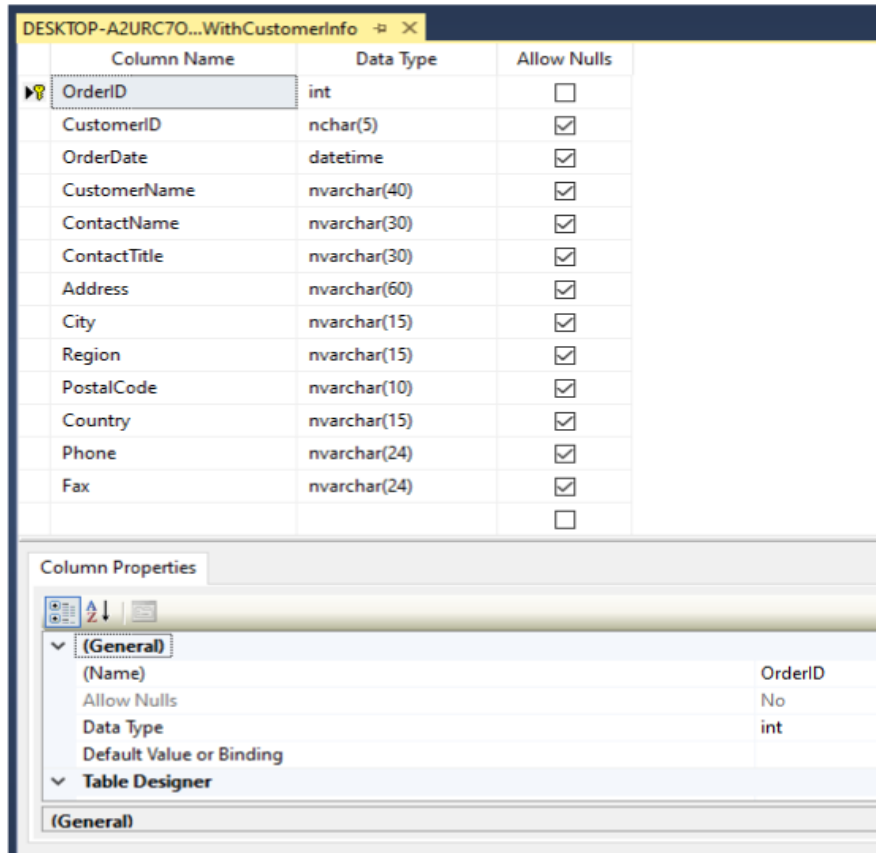
شما می‌توانید یک جدول پیچیده را به‌همراه Join ایجاد کنید. این فرآیند، ساخت چند جدول و تعریف روابط آن‌ها از طریق کلید خارجی را شامل می‌شود. در مثال زیر می‌خواهیم جدولی ایجاد کنیم که اطلاعات دو جدول Orders و Customers از جدول Northwind را ترکیب می‌کند. با جوین این دو جدول، خلاصه‌ای از سفارشات به‌همراه اطلاعات مشتریان به وجود می‌آید.

```
USE Northwind;
GO
```

```
-- Create a complex table combining Orders and Customers information
```

```
CREATE TABLE OrderSummaryWithCustomerInfo (
    OrderID int PRIMARY KEY,
    CustomerID nchar(5),
    OrderDate datetime,
    CustomerName nvarchar(40),
    ContactName nvarchar(30),
    ContactTitle nvarchar(30),
    Address nvarchar(60),
    City nvarchar(15),
    Region nvarchar(15),
    PostalCode nvarchar(10),
    Country nvarchar(15),
    Phone nvarchar(24),
    Fax nvarchar(24),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

اگر به قسمت دیزاین مربوط به جداول پایگاه داده رجوع کنید، متوجه خواهید شد که این جدول دقیقاً با ستون‌ها و شرایط فوق ایجاد شده است.





## ساخت جدول به همراه کلید اصلی ترکیبی

یکی دیگر از موارد استفاده از دستور Create Table در SQL Server، ایجاد جدول همراه با کلید اصلی از نوع Composite است. در مثال زیر، یک جدول تحت عنوان ProductCategories با کلید اصلی ترکیبی ایجاد می‌کنیم.

```
CREATE TABLE ProductCategories (
    ProductID int NOT NULL,
    CategoryID int NOT NULL,
    PRIMARY KEY (ProductID, CategoryID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
```

با اجرای کوئری فوق، محصولات (Products) از طریق کلید اصلی ترکیبی، به چندین Category لینک می‌شوند؛ به طوری که طبقه‌بندی آن‌ها تسهیل بیابد. مشابه تصویر زیر، اکنون جدول ساخته شده باید یک کلید اصلی Composite متشکل از ProductID و CategoryID داشته باشد. البته باید توجه کرد که شما با استفاده از دستور Create Table تنها ساختار جدول را ایجاد خواهید کرد و این فرآیند لزوماً به معنای Insert در SQL Server نیست.

در این بخش از مقاله، نحوه کار با دستور Create Table در SQL Server و ایجاد جدول با آن را به همراه چند مثال بررسی کردیم. پس از ایجاد جدول در SQL Server، ممکن است چندی بعد بخواهید ساختار آن (مثلاً نام یک ستون) را تغییر دهید. در ادامه، نحوه اعمال چنین تغییراتی شرح داده خواهد شد.

## تغییر نام ستون یا جدول در SQL Server

برای تغییر نام ستون یا جدول در SQL Server می‌توان از دستور Alter Table استفاده کرد. ضمن اینکه با استفاده از دستور Alter Table، می‌توان اموری همچون درج ستون‌های جدید، ویرایش نوع‌های داده ستون، Drop کردن ستون‌ها، درج / تغییر محدودیت‌ها، درج Computed Column، پارتیشن‌سازی و [ایندکس گذاری](#) (Indexing) انجام داد.

## درج ستون جدید به جدول

در مثال زیر، ستون Phone Number را به همراه مقدار پیش‌فرض به جدول Customer درج می‌کنیم.

```
ALTER TABLE Customers
ADD PhoneNumber varchar(20) NOT NULL DEFAULT 'N/A';
```

با اجرای کوئری بالا، ستون Phone Number درج می‌شود؛ این ستون نمی‌تواند Null باشد و مقدار N/A به عنوان پیش‌فرض برای آن در نظر گرفته شده است.

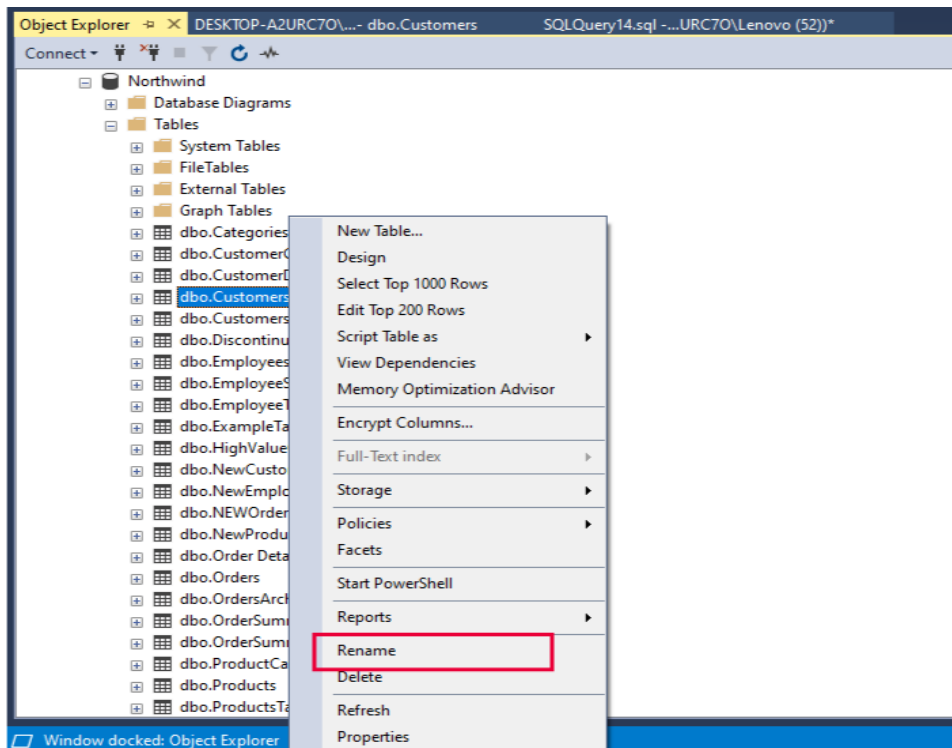
CustomerID	CompanyName	PhoneNumber	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone
1	ALFK	N/A	Maria Anders	NULL	NULL	NULL	NULL	NULL	Germany	NUL
2	ALFKI	N/A	Maria Anders	Sales Representative	Obere Str. 57	Seattle	NULL	12209	Germany	030-
3	ANATR	N/A	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	NULL	05021	Mexico	(5) 5-
4	ANTON	N/A	Antonio Moreno	Owner	Mataderos 2312	México D.F.	NULL	05023	Mexico	(5) 5-
5	AROUT	N/A	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	NULL	WA1 1DP	UK	(171)
6	AZBRE	N/A	Sarah Colon	NULL	NULL	NULL	NULL	NULL	France	NUL
7	BERGS	N/A	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	NULL	S-958 22	Sweden	0921
8	BLAUS	N/A	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	NULL	68306	Germany	0621
9	BLONP	N/A	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	NULL	67000	France	88.61
10	BOLID	N/A	Martín Sommer	Owner	C/ Araquil, 67	Madrid	NULL	28023	Spain	(91) 1-

## تغییر نام جدول از طریق SSMS

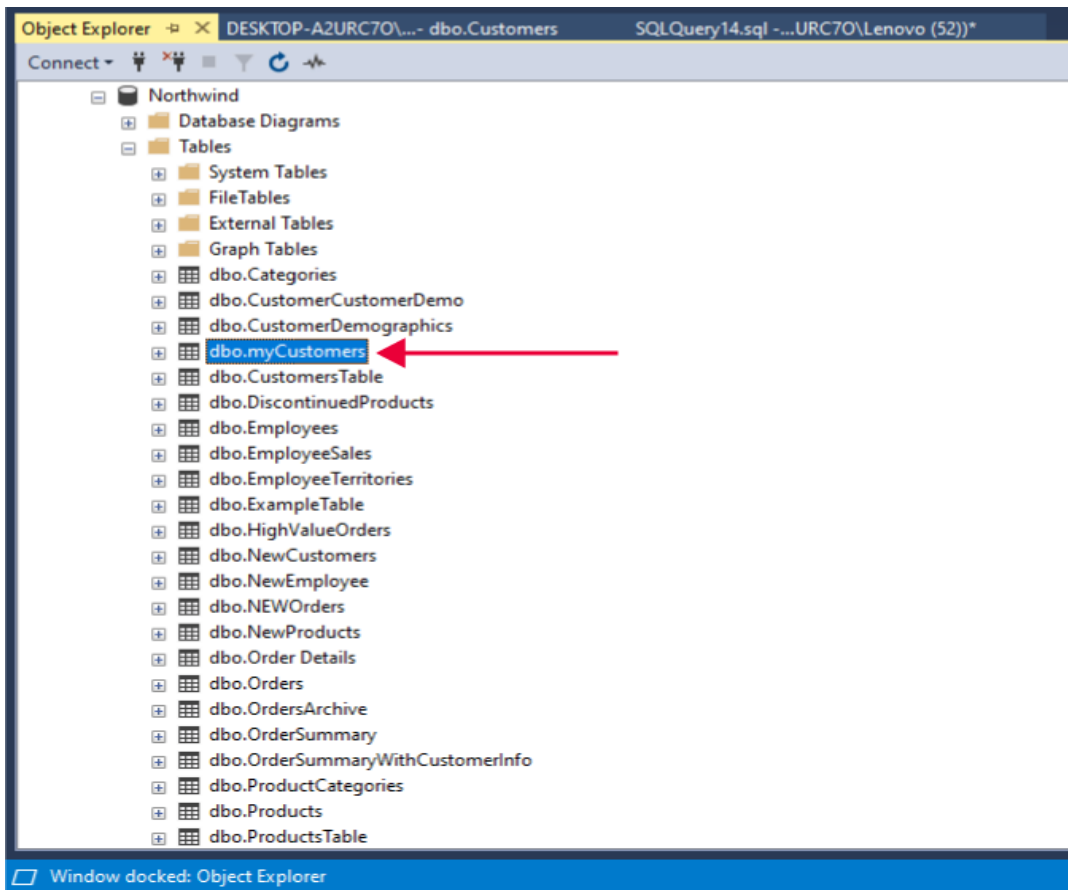
شما می‌توانید نام جداول را از طریق دستور Alter Table در SQL Server تغییر دهید؛ با این وجود، تغییر آن از طریق SSMS آسان‌تر است. برای این کار، کفایت مراحل زیر را دنبال کنید:

فرض کنید می‌خواهیم نام جدول Customer را به myCustomer تغییر دهیم.

۱. از بخش Object Explorer، روی جدول موردنظر، کلیک راست کرده و گزینه Rename را انتخاب کنید.



۲. نام جدید جدول را وارد و پس از آن، روی Enter کلیک کنید.

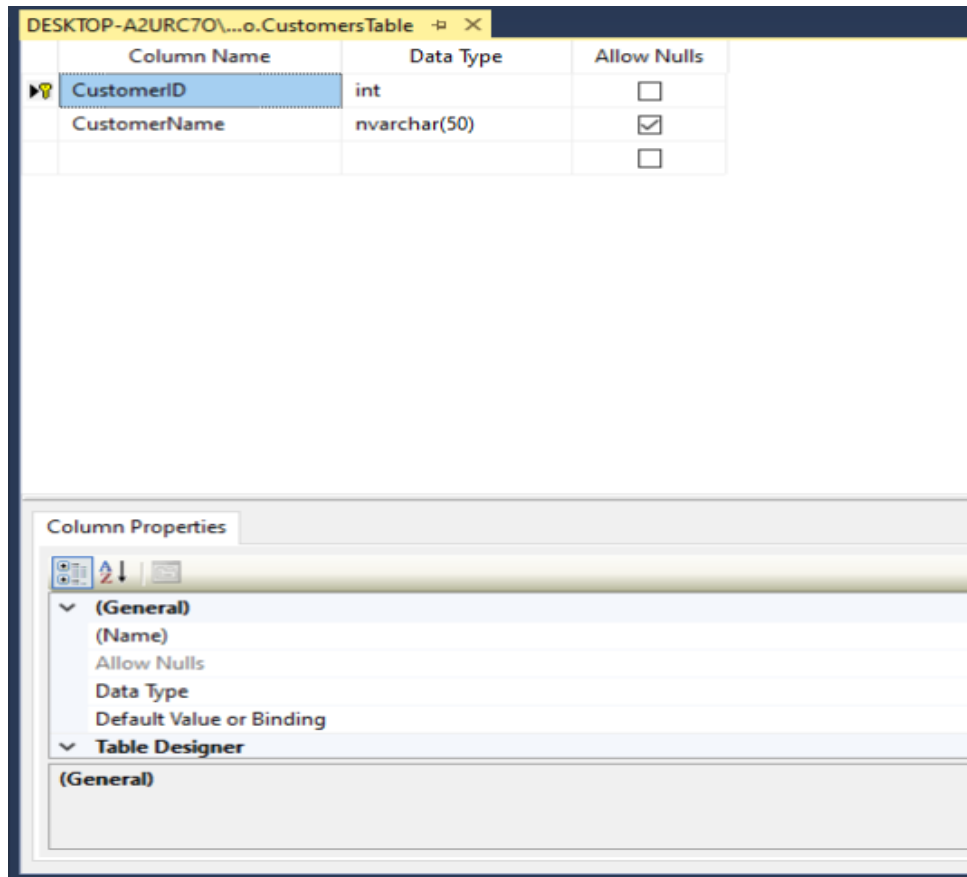


### حذف ستون های جدول در SQL Server

برای حذف ستون‌هایی از جدول در SQL Server، لازم است از دستور Alter Table به همراه Drop Column استفاده شود. در مثال زیر، ستون Email را از جدول CustomersTable حذف خواهیم کرد.

```
ALTER TABLE CustomersTable
DROP COLUMN Email
```

با اجرای کوئری بالا، ستون مذکور از جدول مدنظر حذف خواهد شد و نمای جدول به صورت زیر خواهد بود:

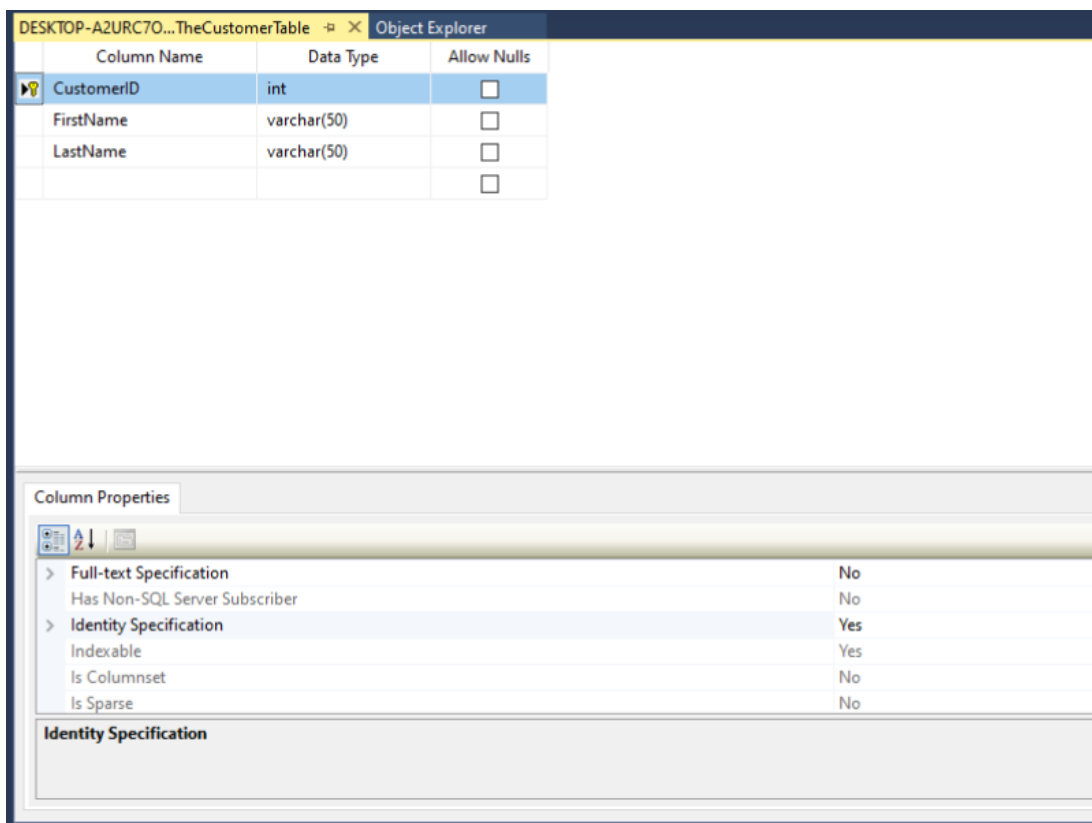


### آشنایی با مفهوم Auto Increment در SQL Server

مفهوم Auto Increment در SQL Server ، قابلیت است که به کمک آن، امکان تولید خودکار مقادیر یکتا (Unique) برای یک ستون خاص از جدول در زمان درج رکورد جدید فراهم می‌شود. این مشخصه به‌طور خاص برای مواقعی کاربردی است که می‌خواهید شناسه‌گرهای خاص و کلید اصلی ایجاد کنید. رایج‌ترین روش پیاده‌سازی Auto Increment هنگام ایجاد جدول در SQL Server است. برای این‌کار، کفایت ستون موردنظر را به‌عنوان ستون Identity تعریف کنید. شایان ذکر است که می‌توان این قابلیت را از طریق تریگرها و Sequence ها نیز اعمال کرد. برای درک بهتر، به مثال زیر توجه کنید:

```
CREATE TABLE TheCustomerTable (
    CustomerID int IDENTITY(1, 1) PRIMARY KEY,
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL
);
```

با اجرای کوئری فوق، یک جدول به نام TheCustomerTable ایجاد می‌شود که در آن، ستون CustomerID کلید اصلی جدول و به صورت Auto Incremental است. در این جدول، دو ستون دیگر برای نام و نام خانوادگی مشتریان ایجاد می‌شود؛ توجه کنید که مقادیر این دو ستون نمی‌توانند Null باشند. به این ترتیب، هر مشتری جدیدی که در جدول درج شود، یک ID یکتا و منحصر به فرد دریافت می‌کند.



### کلام آخر: کاربرد دستور Create Table در SQL Server چیست؟

دستور Create Table در SQL Server به شما قابلیت‌های همچون تعریف اسکیمای و ساخت جدول را می‌دهد و می‌توانید از طریق آن، ستون‌ها، کلید اصلی و محدودیت‌های مورد نظر را روی جدول تعریف کنید. در این مقاله به چگونگی ایجاد جدول در SQL Server پرداخته شد. برای آشنایی بیشتر با سایر دستورات T-SQL، پیشنهاد می‌شود [مقاله معرفی پرکاربردترین دستورات SQL Server](#) را نیز مطالعه کنید.