



عنوان مقاله: Schema در SQL Server چیست؟ آشنایی با روش های ایجاد اسکیمای

نویسنده مقاله: تیم فنی نیک آموز

تاریخ انتشار: ۲۷ اسفند ۱۴۰۲

منبع: <https://nikamooz.com/schema-in-sql-server/>

Schema در SQL Server نظارت بر پایگاه داده را بهبود می بخشد و در کاهش ناسازگاری و مدیریت کارآمد اشیای دیتابیس مفید واقع می شود. به عنوان [سیستم مدیریت پایگاه داده رابطه ای](#) (RDBMS)، نرم افزار SQL Server به منظور مدیریت، سازماندهی و استخراج حجم وسیعی از داده، مورد استفاده قرار می گیرد و شما می توانید به واسطه اجرای [دستور Insert](#) و [دستور آپدیت](#)، همچنین [دستور Select](#) و [عبارت Where](#) و سایر موارد، این امور را انجام دهید. در این مطلب، به بررسی چگونگی استفاده از اسکیمای در SQL Server می پردازیم و آن را به زبان قابل درک شرح می دهیم. برای درک عمیق تر مفاهیم SQL Server، پیشنهاد می کنیم [آموزش جامع SQL Server](#) را مطالعه کنید. علاوه بر این، برای یادگیری نحوه خودکارسازی تسک های تکرارشونده و تسریع فرآیندها، به [مقاله اسکریپت های پرکاربرد SQL](#) رجوع کرده و آن را مطالعه کنید.

Schema در SQL Server چیست؟

Schema در SQL Server نوعی کانتینر است که مجموعه هایی از اشیای منطقی پایگاه داده (Database) را گروه بندی می کند. ممکن است این اشیا انواع جداول، View ها، [انواع استور پروسیجر](#) (Stored Procedures)، توابع، ایندکس ها و [انواع تریگرها](#) باشند. برای آشنایی بیشتر با این اجزا، پیشنهاد می شود [مقاله بررسی نحوه ایجاد جدول و روش ایجاد View در SQL Server](#) و همچنین، [انواع ایندکس گذاری](#) را مطالعه کنید. درحقیقت، می توان Schema را مانند راهی در نظر داشت که پایگاه های داده را به فولدرهای مشخصی سازماندهی می کند. وجود اسکیمای در SQL Server، به بهبود شفافیت و نگهداری دیتابیس ها، افزایش امنیت و جلوگیری از ناسازگاری های نام گذاری منجر می شود.

کاربرد Schema در SQL Server

در این بخش، چند مورد از کاربردهای رایج اسکیمای در SQL Server بررسی می شوند.

- **اپلیکیشن های Multi-Tenant:** اگر پایگاه داده به گونه ای است که به چند مشتری و سازمان خدمات ارائه می دهد، می توان از اسکیمای جداگانه برای جداسازی داده های هر Tenant استفاده کرد. این اقدام، امنیت و حریم خصوصی داده ها را بهبود خواهد داد.
- **محیط های توسعه، تست و تولید:** می توانید برای هر محیط (توسعه، تست، تولید) اسکیمای جداگانه ایجاد کنید تا تغییرات توسعه به صورت ایزوله نگهداری شوند و از Conflict با داده های فرآیند تولید اجتناب شود.
- **جداسازی منطقی عملکردها:** اگر پایگاه داده شما از قابلیت ها یا ماژول های مختلف پشتیبانی می کند، می توانید جداول، ویوها و رویه های مرتبط را برای سازماندهی بهتر در اسکیمای جداگانه گروه بندی کنید.

- **داده‌های مخصوص اپلیکیشن:** ممکن است اپلیکیشن‌های مختلفی با پایگاه داده تعامل داشته باشند. استفاده از اسکیمای جداگانه برای داده‌های هر برنامه، کنترل دسترسی را تسهیل می‌دهد و خطر تغییرات تصادفی را کاهش خواهد داد.
 - **کنترل دسترسی کاربر:** با اختصاص دادن مجوز به انواع Schema در SQL Server، می‌توانید براساس نیازها، به کاربران دسترسی‌هایی به مجموعه‌ای از اشیا پایگاه داده اعطا کنید.
- در مجموع، به‌کارگیری Schema در SQL Server، سازماندهی بهبودیافته، ایمن‌سازی و مدیریت کارآمدتر پایگاه داده را به همراه دارد. این موضوع به‌طور خاص، برای رسیدگی به ساختار داده‌های پیچیده یا کاربران متعدد صادق است.

مزایای ایجاد Schema در SQL Server

مزیت‌های استفاده از Schema در SQL Server عبارتند از:

- گروه‌بندی منطقی اشیا دیتابیس، تسهیل درک و جابه‌جایی میان پایگاه داده
- بهبود ساختاردهی و مدیریت شی‌های مختلف در اسکیمایها
- وجود **دسترسی کنترل دانه دانه** (Granular Access Control)
- تسهیل مدیریت اشیا
- قابلیت ایزوله‌سازی کاربران

حال این سؤال پیش می‌آید که ایجاد Schema در SQL Server به چه صورت است؟

روش های ایجاد Schema در SQL Server

در این بخش به تشریح اصلی‌ترین روش‌های ایجاد Schema در SQL Server می‌پردازیم.

ایجاد Schema با T-SQL

با فرض اینکه شما از قبل، **نصب SSMS** را انجام داده‌اید و به آن دسترسی دارید، SQL Server Management Studio را اجرا کنید. در این مطلب، از **دیتابیس Northwind** به‌منظور اجرای کوئری‌ها استفاده می‌شود.

فرض کنید می‌خواهیم منطق سفارش‌ها و مشتریان را از یکدیگر جدا کنیم تا سازماندهی آن‌ها به شکل بهبودیافته‌تری انجام شده و به‌طور کلی، امنیت بیشتری حاصل شود.

۱- ایجاد اسکیمای

روش اول برای ایجاد Schema در SQL Server، اجرای کوئری زیر است:

```
USE northwind;
GO
CREATE SCHEMA CustomerData;
GO
CREATE SCHEMA OrderData;
```

با اجرای کوئری فوق، یک اسکیمای برای داده‌ها مشتریان و یکی برای داده‌ها سفارش‌ها ایجاد می‌شود. توجه کنید که برای جلوگیری از خطا یا مشکلات احتمالی، از جداساز GO به منظور اتمام Batch ها استفاده شده است. برای آن که بررسی کنید آیا ساخت اسکیمای در SQL Server به درستی انجام شده یا خیر، می‌توانید با اجرای کوئری زیر، لیست تمامی اسکیمای موجود در دیتابیس Northwind را مشاهده کنید:

```
USE northwind;
SELECT * FROM sys.schemas;
```

پس از اجرای این کوئری، خروجی مشابه زیر مشاهده خواهید کرد.

	name	schema_id	principal_id
1	dbo	1	1
2	guest	2	2
3	INFORMATION_SCHEMA	3	3
4	sys	4	4
5	Sales	5	1
6	CustomerData	6	1
7	OrderData	7	1
8	db_owner	16384	16384
9	db_accessadmin	16385	16385
10	db_securityadmin	16386	16386
11	db_ddladmin	16387	16387

۲- انتقال جداول مربوطه به Schema ها

برای انتقال جداول مرتبط به Schema های ایجادشده، از دستور Alter Schema استفاده می‌شود. به کوئری زیر توجه کنید:

```
ALTER SCHEMA CustomerData TRANSFER dbo.Customers;
ALTER SCHEMA OrderData TRANSFER dbo.Orders;
```

توجه: اگر اسکیمای شما نام دیگر به غیر از dbo دارا است، آن را به جای dbo در کوئری فوق قرار دهید. برای بررسی اینکه آیا جداول مربوطه به اسکیمای منتقل شده اند یا خیر، می توانید کوئری پایین را اجرا کنید.

```
SELECT *
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'CustomerData' AND TABLE_NAME = 'Customers';
```

```
SELECT *
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'OrderData' AND TABLE_NAME = 'Orders';
```

The screenshot shows a SQL query window with two queries. The first query is:

```
SELECT *
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'CustomerData' AND TABLE_NAME = 'Customers';
```

The second query is:

```
SELECT *
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'OrderData' AND TABLE_NAME = 'Orders';
```

Below the queries, the Results pane shows two tables. The first table has the following data:

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
Northwind	CustomerData	Customers	BASE TABLE

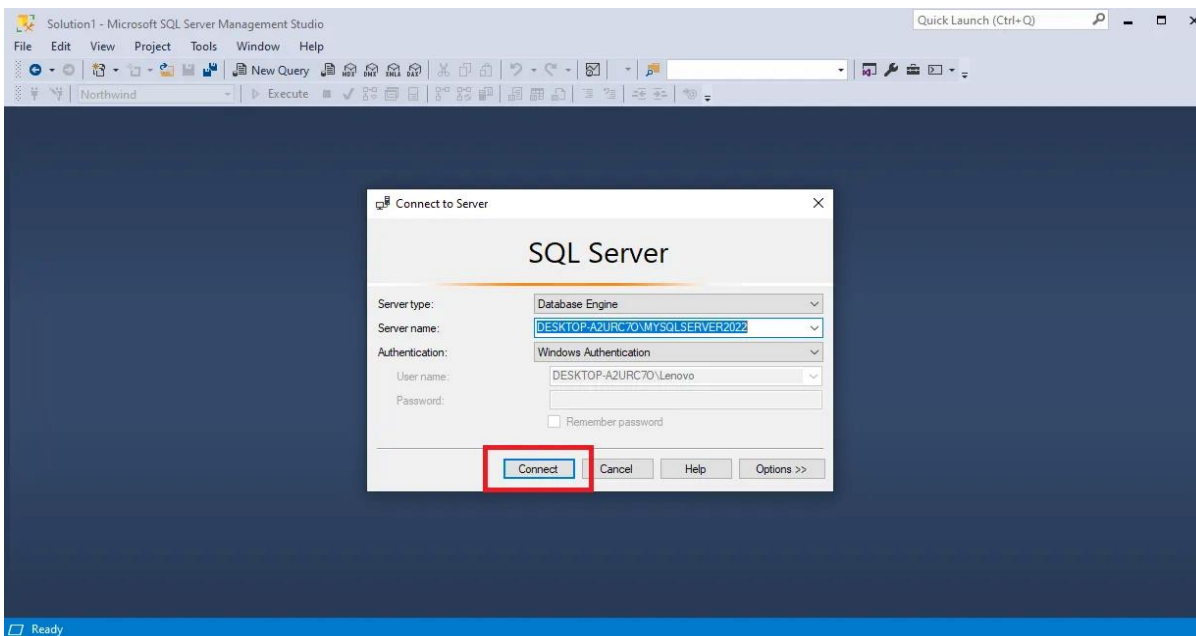
The second table has the following data:

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
Northwind	OrderData	Orders	BASE TABLE

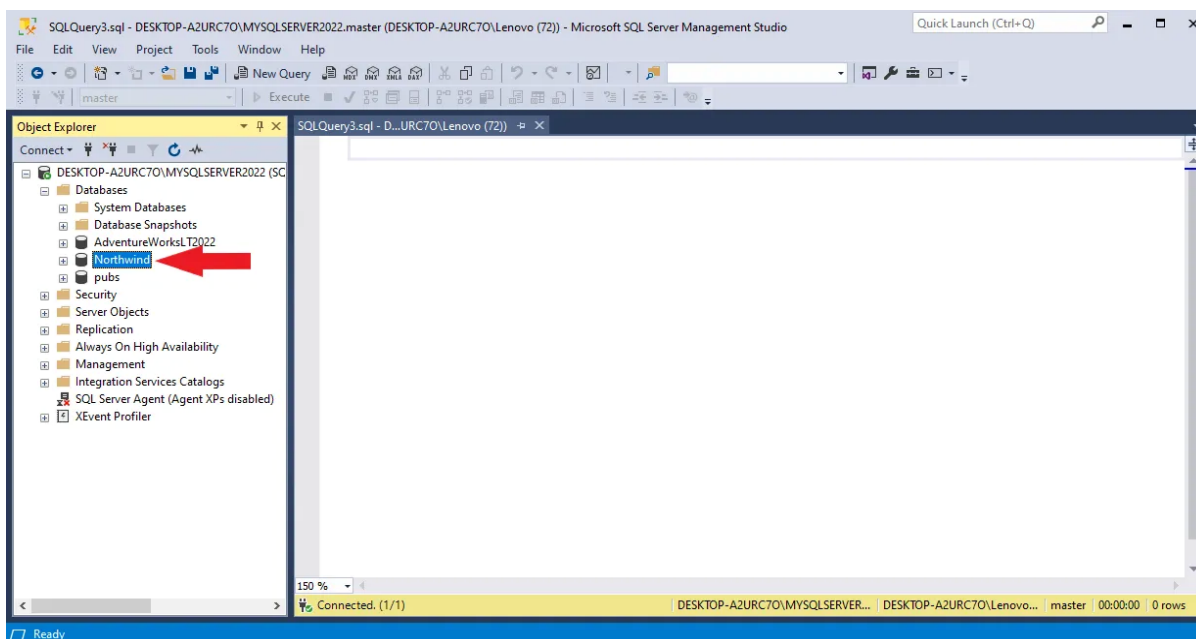
Red arrows point to the 'BASE TABLE' column in both result rows.

ایجاد Schema با SSMS

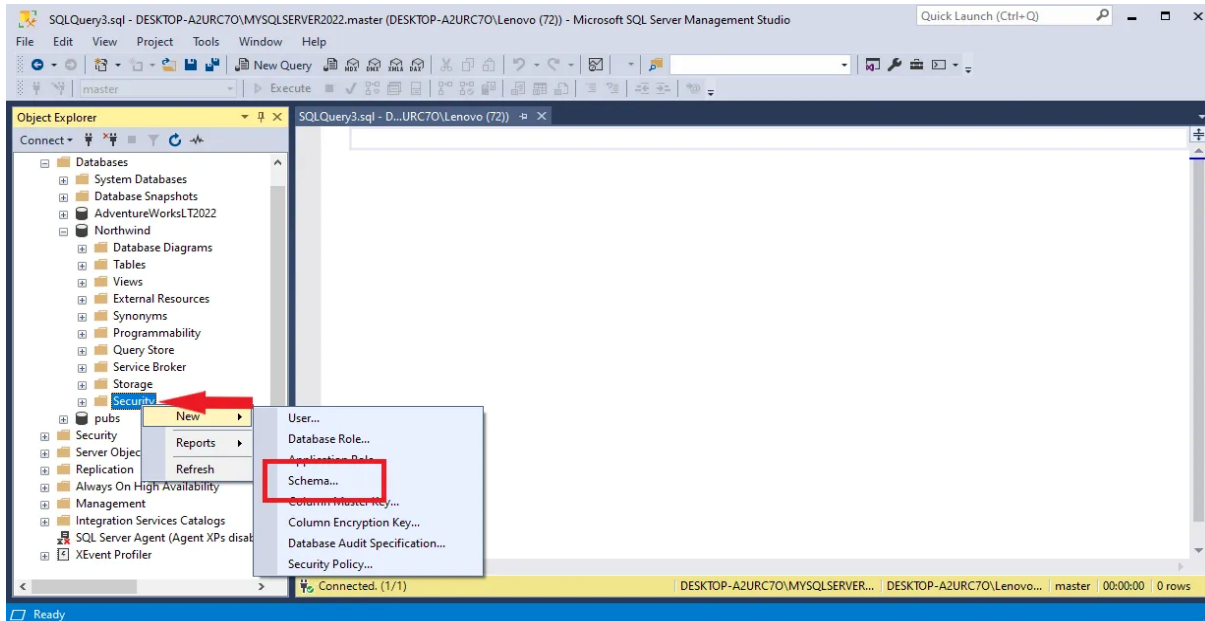
روش دوم برای ایجاد Schema در SQL Server ، استفاده از رابط گرافیکی است که در این بخش بررسی خواهد شد. ۱. در ابتدا لازم است SMSS را روی سیستم اجرا کرده و به نمونه SQL Server خود Connect شوید.



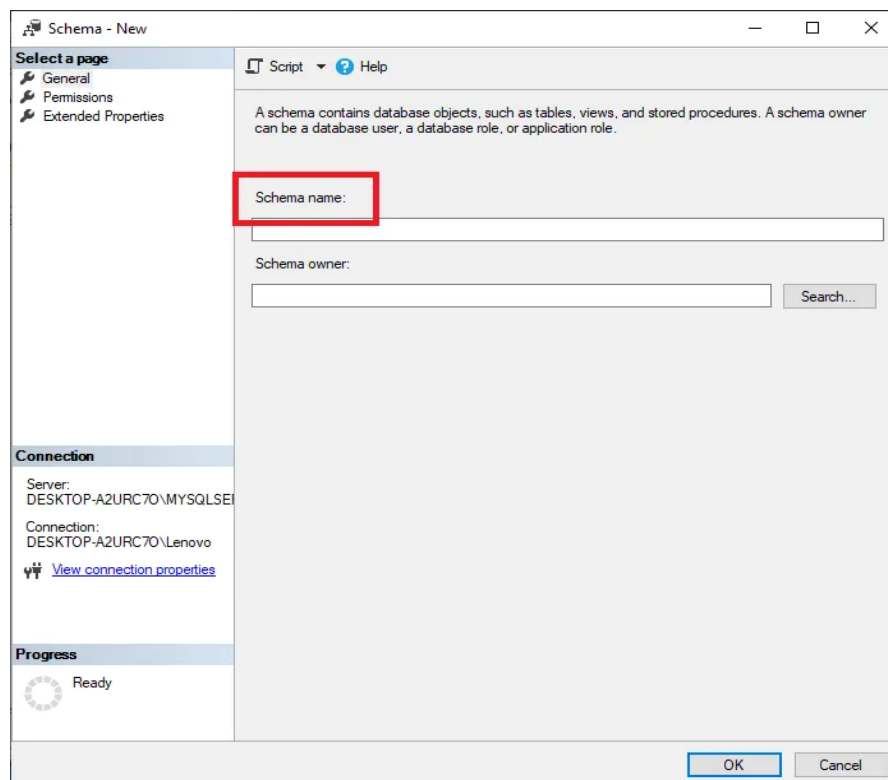
۲. از بخش Object Explorer، روی فولدر Databases کلیک کرده و سپس، آن دیتابیس‌ها را انتخاب کنید که می‌خواهید برای آن اسکیمای ایجاد کنید را انتخاب نمایید.



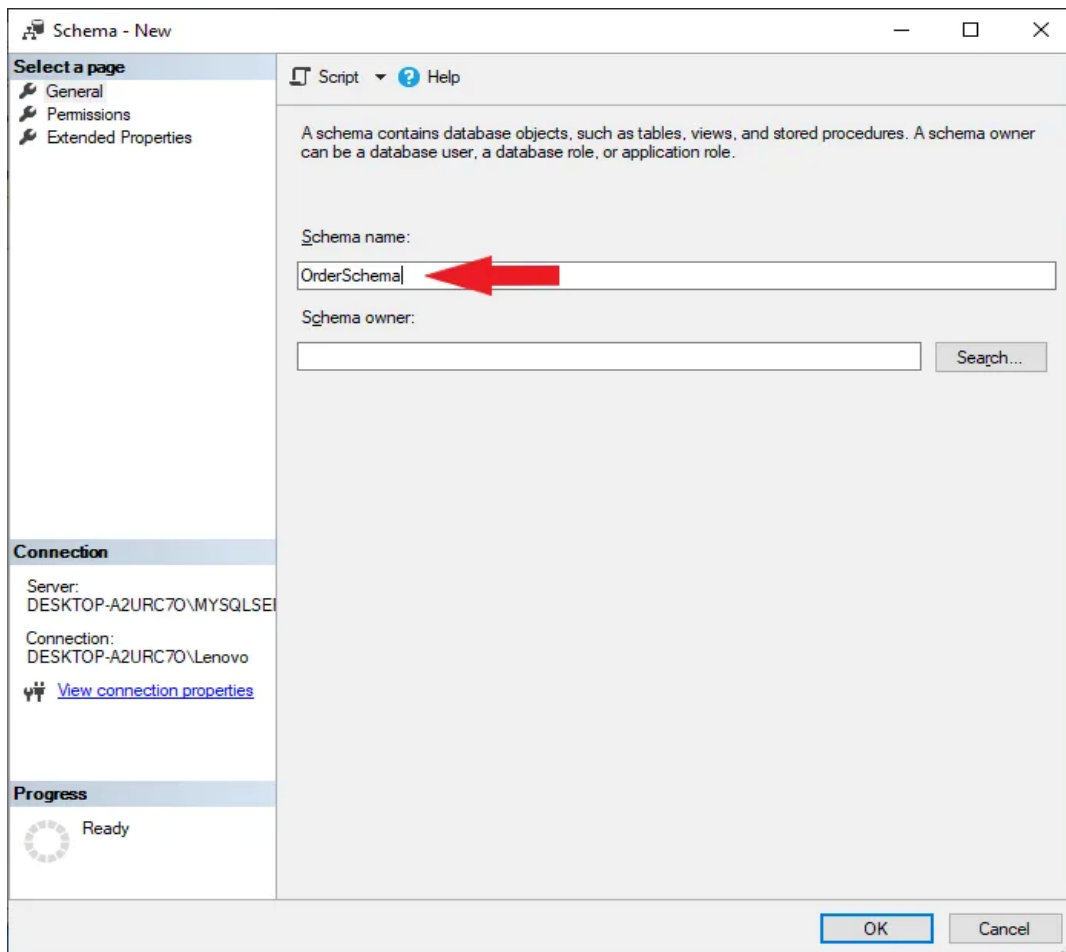
۳. روی فولدر Security از پایگاه داده مدنظر کلیک راست کنید، گزینه New و سپس گزینه Schema را از منو انتخاب کنید.



۴. در پنجره New - Schema، یک فیلد متنی برای نام اسکیمای در SQL Server و یک فیلد برای تعیین مالک اسکیمای (Schema Owner) قرار دارد. در بخش Schema name، نام اسکیمای موردنظر خود را وارد کنید. توجه کنید که وارد کردن Schema Owner انتخابی است. ضمن اینکه در این بخش امکان جستجوی کاربران یا Role ها از طریق دکمه Search وجود دارد.



فرض کنید بخواهیم یک اسکیمای تحت عنوان OrderSchema ایجاد کنیم. این نام را در بخش Schema Name وارد کرده و روی دکمه OK کلیک کنید.



۵. اکنون می‌خواهیم بررسی کنیم آیا Schema موردنظر به‌درستی در پایگاه داده ایجاد شده است یا خیر. برای این کار، کوئری زیر را اجرا کنید:

```
SELECT * FROM sys.schemas;
```

همان‌طور که در تصویر مشاهده می‌کنید، Schema مدنظر با نام OrderSchema و از طریق SSMS ایجاد شده است.

SQLQuery3.sql - D:\URC70\Lenovo (72)*

```
SELECT * FROM sys.schemas;
```

150 %

Results Messages

	name	schema_id	principal_id
1	dbo	1	1
2	guest	2	2
3	INFORMATION_SCHEMA	3	3
4	sys	4	4
5	Sales	5	1
6	CustomerData	6	1
7	OrderData	7	1
8	OrderSchema	8	1
9	db_owner	16384	16384

نکته حائز اهمیت این است که برای ساخت Schema در SQL Server، دارا بودن پرمیشن Create Schema روی آن دیتابیس ضروری است.

تغییر Schema در SQL Server

برای تغییر Schema در SQL Server، دو راه پیش روی شما قرار دارد که در ادامه به آن‌ها می‌پردازیم. اما پیش از آن، توصیه می‌شود [ویدئو Schema Binding و تاثیر آن بر کارایی Function های SQL Server](#) را نیز مشاهده کنید تا شناخت بهتری از مفاهیم پراهمیت مربوط به اسکیمای به دست آورید.

روش اول: استفاده از عبارات T-SQL

فرض کنید می‌خواهیم دو اسکیمای ایجادشده در بخش پیشین را تغییر دهیم. در چنین شرایطی مراحل زیر را دنبال کنید.

- ساخت اسکیمای جدید: با اجرای کوئری زیر، Schema هایی با نام‌های جدید ایجاد کنید:

```
CREATE SCHEMA NewCustomerData;
Go
CREATE SCHEMA NewOrderData;
```


- انتقال جداول به اسکیماهای جدید: برای انتقال جدول ها، کوئری زیر را Run کنید:

```
ALTER SCHEMA NewCustomerData TRANSFER dbo.Customers;
```

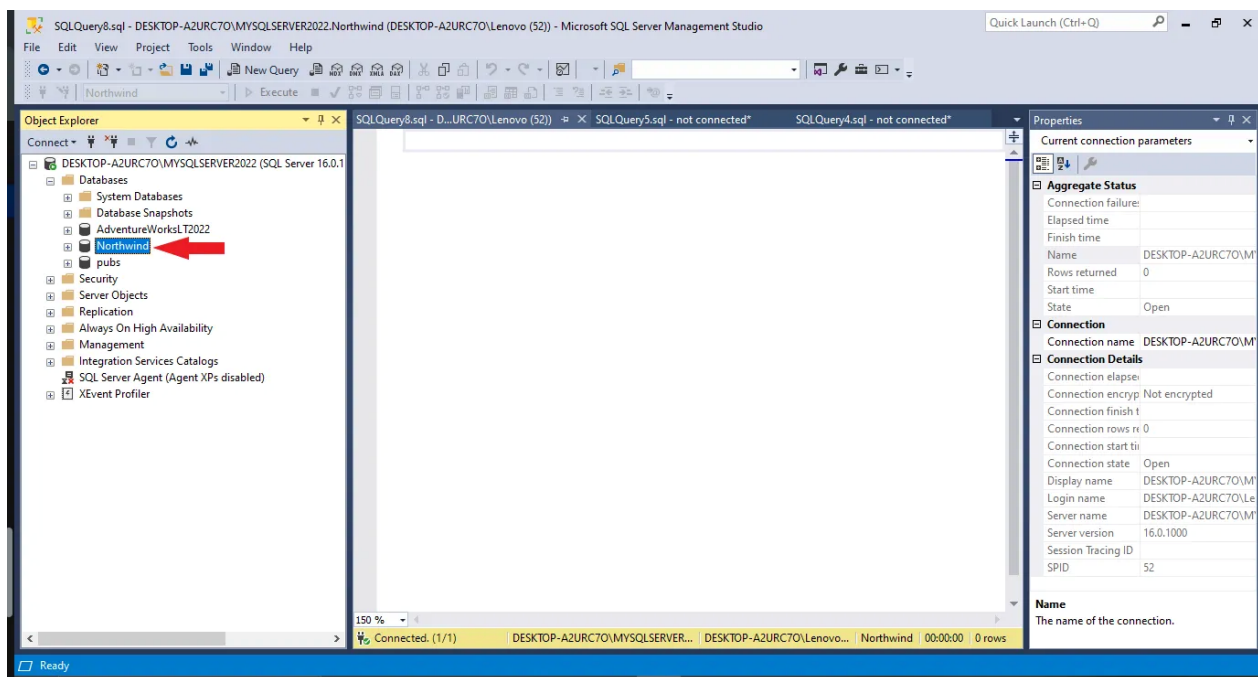
```
ALTER SCHEMA NewOrderData TRANSFER dbo.Orders;
```

به این ترتیب، شما به تغییر اسکیمای در SQL Server پرداخته‌اید.

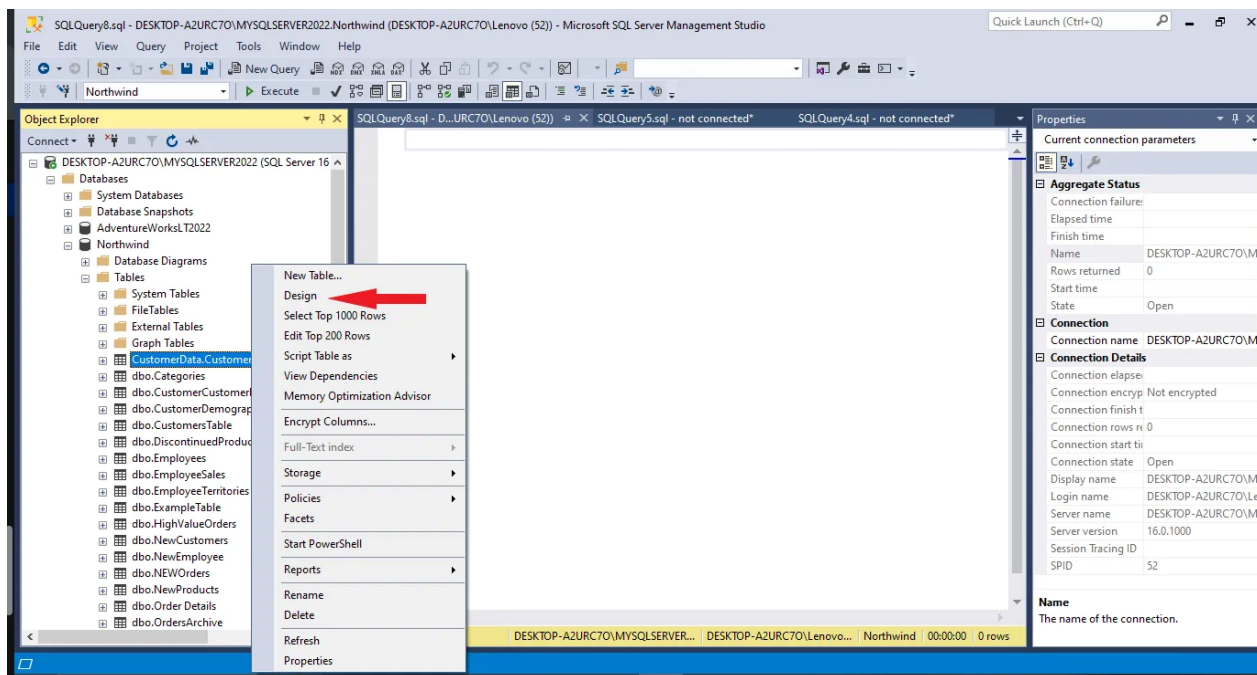
روش دوم: استفاده از SSMS

شما می‌توانید مشابه ساختن Schema در SQL Server از طریق SSMS، برای ویرایش آن‌ها اقدام کنید. با دنبال کردن مراحل زیر، می‌توانید اسکیمای را تغییر دهید.

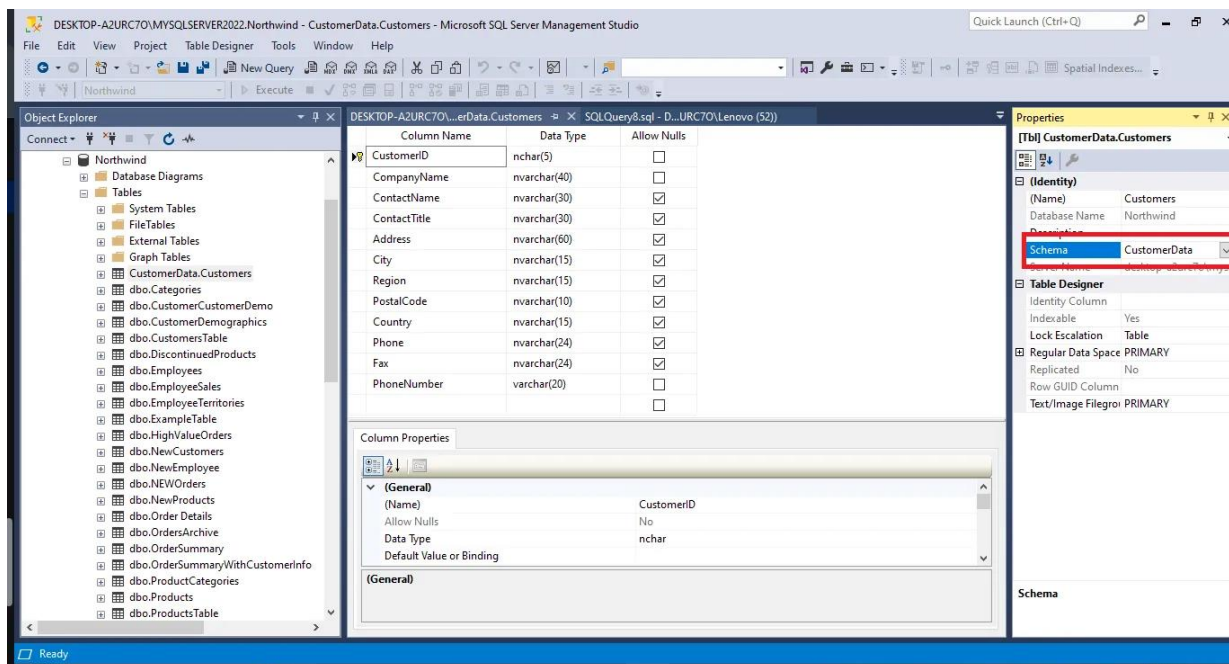
- یافتن جدول موردنظر: از بخش Object Explorer، پایگاه داده‌ای را انتخاب کنید که قصد تغییر اسکیمای آن را دارید. سپس، روی فولدر Tables از دیتابیس کلیک کنید تا Expand شود.



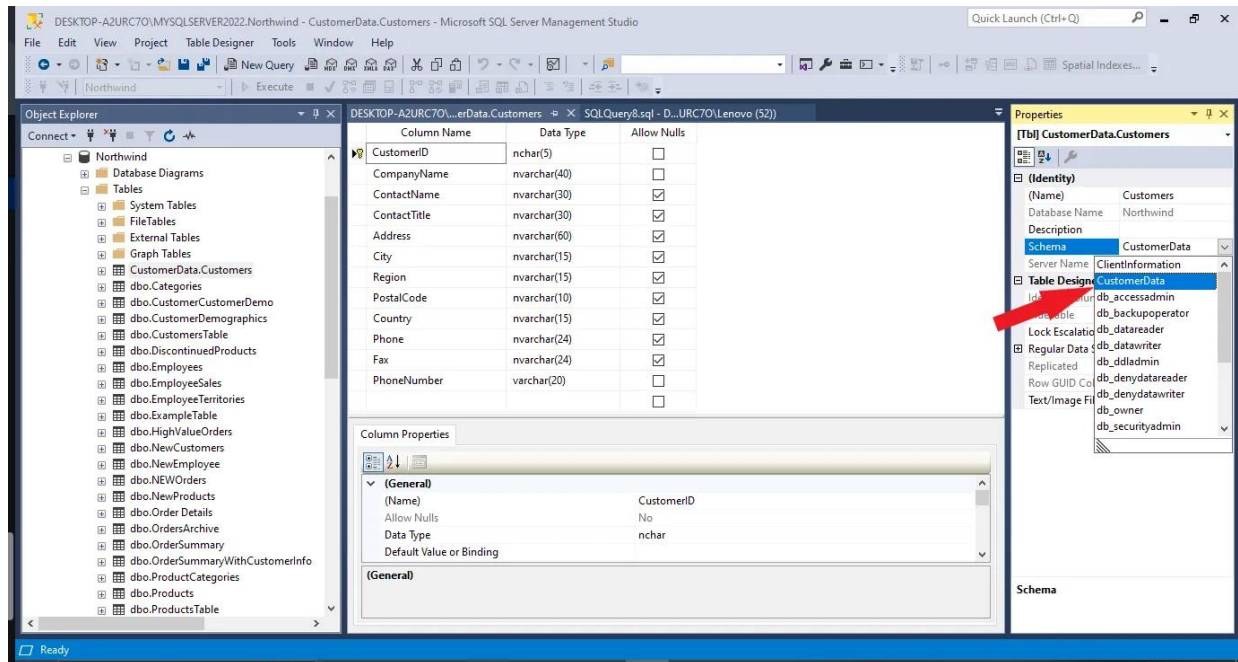
- دسترسی به Properties جدول: روی جدول مربوطه کلیک راست کرده و گزینه Design را انتخاب کنید.



- تغییر مالکیت اسکیمای در SQL Server: پس از ورود به پنجره Design، پنجره Properties را در سمت راست صفحه نمایش پیدا کنید. اگر این پنل در سمت راست صفحه نیست، لازم است دکمه F4 را فشار دهید تا Properties روی صفحه نمایش نمایان شود.



- انتخاب یک اسکیمای جدید: روی منوی Dropdown کنار Schema کلیک کرده و اسکیمای جدید موردنظر خود را از آن انتخاب کنید.



- ذخیره سازی تغییرات: زمانی که اسکیمای جدید را انتخاب کردید، می توانید پنجره Properties را ببندید. با استفاده از Ctrl+S، تغییرات را ذخیره سازی کنید.
- توجه به ملاحظات جانبی: باید به این موضوع توجه کرد که ممکن است تغییر مالکیت یک اسکیمای در SQL Server، آپدیت اشیا را نیاز داشته باشد که به جدول ارجاع داده شده باشند. علاوه بر این، پیشنهاد می شود پیش از پیاده سازی چنین تغییراتی در محیط محصول، آن ها را در محیط توسعه تست کنید.

حذف Schema در SQL Server

حذف Schema در SQL Server به صورت مستقیم امکان پذیر نیست و شما باید اسکیمای موردنظر را با استفاده از عبارت T-SQL، اصطلاحاً دراپ (Drop) کنید. به منظور حذف اسکیمای، اقدامات زیر را انجام دهید:

- اطمینان از خالی بودن اسکیمای: تنها در شرایطی می توان اسکیمای را حذف کرد که در آن هیچ اشیا، مانند جدول، ویو، استور پروسیجر و سایر موارد وجود نداشته باشد. بنابراین، شما باید اشیا موجود در اسکیمای را از طریق یکی از روش های زیر حذف کنید:
 - حذف دستی اشیا: روی هر شی کلیک راست کرده و گزینه Delete را انتخاب کنید.
 - حذف مبتنی بر اسکریپت: براساس نوع شی، می توان با استفاده عبارات Drop Table، Drop View و... شی ها را حذف کرد.

۲. **استفاده از عبارت DROP SCHEMA**: پس از اطمینان از خالی بودن اسکیمای، یک پنجره Query جدید در SSMS باز کرده و با کمک ساختار زیر آن را Drop کنید:

```
DROP SCHEMA [schema_name]
```

قسمت [schema_name] را با نام اصلی اسکیمای خود جایگزین کنید.

۳. **بررسی اشیای فعلی (انتخابی)**: اگر خیالتان از وجود اسکیمای راحت نیست، می‌توانید از عبارت IF EXIST استفاده کنید تا از خطاهای احتمالی جلوگیری شود.

```
DROP SCHEMA IF EXISTS [schema_name];
```

توجه کنید که باید مجوزهای لازم برای Drop کردن اسکیمای در SQL Server را داشته باشید. این موضوع از طریق CONTROL در اسکیمای یا ALTER ANY SCHEMA روی دیتابیس امکان‌پذیر است. علاوه بر این، دقت کنید که با حذف اسکیمای از طریق Drop، بازگشت از این عمل ممکن نیست. به همین دلیل، بهتر است دراپ را در محیطی غیر از محیط محصول اعمال کنید تا آسیبی به دیتابیس اصلی وارد نشود.

نحوه مشاهده Schema در SQL Server

برای مشاهده Schema در SQL Server، از ساختار کلی زیر استفاده کنید:

```
SELECT name  
FROM sys.schemas;
```

اگر می‌خواهید این بخش را به همراه مثال قابل درک متوجه شوید، [مقاله لیست Schema های موجود در یک دیتابیس](#) و همچنین، [مقاله لیست Schema های ایجاد شده توسط کاربر در یک دیتابیس](#) می‌توانند برایتان مفید باشند.

سخن پایانی: چرا اسکیمای در SQL Server پراهمیت است؟

به واسطه وجود اسکیمای در SQL Server، امکان مدیریت شی‌های پایگاه داده، از جدول گرفته تا استور پروسجرها، وجود دارد. در عمل، Schema با گروه‌بندی اشیای براساس کارکرد یا مالکیت، سازماندهی دیتابیس را بهبود می‌بخشد و درک ساختار پایگاه داده را ساده‌سازی می‌کند. ضمن اینکه با کمک آن، به نوعی امنیت و همچنین، کنترل دسترسی به دیتابیس SQL Server تضمین می‌شود. در این مقاله، به بررسی نحوه ایجاد، تغییر و حذف اسکیمای در SQL Server پرداخته شد تا شما بتوانید از آن به‌عنوان راهنمای قابل درک بهره‌مند شوید.