



عنوان مقاله: تعریف محدودیت (Constraint) در SQL Server + بررسی انواع و کاربردهای آن ها

نویسنده مقاله: تیم فنی نیکآموز

تاریخ انتشار: ۱۵ بهمن ۱۴۰۲

منبع: [/https://nikamooz.com/constraint-definition-in-sql-server](https://nikamooz.com/constraint-definition-in-sql-server)

محدودیت در SQL Server ، نقش کلیدی و بنیادی در حفظ یکپارچه‌سازی و سازگاری پایگاه‌های داده دارد. با اعمال Constraint روی جداول، تضمین می‌شود که داده‌ها از شرایط و قوانین خاصی پیروی می‌کنند. پیش‌تر به بررسی **پربکارترین دستورات SQL Server**، از جمله **دستور Insert** و **دستور Update** و **دستور Select** پرداخته شد. اکنون در این مطلب، انواع Constraint در SQL Server و کاربردهای هر یک در کوئری‌نویسی به‌همراه خروجی تشریح خواهند شد.

محدودیت در SQL Server چیست؟

محدودیت‌ها (Constraints) در SQL Server ، قوانینی محسوب می‌شوند که می‌توان آن‌ها را روی نوع داده‌های یک جدول اعمال کرد؛ این یعنی، به واسطه Constraint ها امکان ایجاد محدودیت خاصی روی نوع داده قابل ذخیره‌سازی در یک ستون مشخص وجود دارد. به این ترتیب، **یکپارچگی** (Integrity) و دقت (Accuracy) داده‌ها تضمین می‌شود و از ورود دیتای ناسازگار به جداول پایگاه داده جلوگیری خواهد شد.



مزایای تعریف محدودیت در SQL Server

تعریف محدودیت در SQL Server ، مزیت‌های زیر را برای پایگاه داده شما به همراه دارد:

- **جلوگیری از درج داده‌های غیرمرتبط:** شما می‌توانید با تعریف کردن محدودیت در SQL Server ، از درج داده‌های نادرست یا ناسازگاری به جدول جلوگیری کنید. بدین طریق، دقت و قابل اکتفا بودن اطلاعات تضمین می‌شود.
 - **اعمال قوانین تجاری:** با استفاده از Constraint ها می‌توان قوانین خاص کسب و کار را تعریف کرد. در چنین شرایطی، مطمئن خواهید شد که داده‌های جداول دیتابیس از قوانین تجاری پیروی می‌کنند. وجود این سطح از سازگاری، برای حفظ یکپارچگی داده‌ها و کاهش خطا حائز اهمیت است.
 - **داده‌های استانداردشده:** با اعمال Constraint در SQL Server ، تطبیق‌پذیری در پایگاه داده تضمین می‌شود و به دنبال آن، روند دستکاری و تحلیل داده‌ها سهولت می‌یابد.
 - **حفظ یکپارچگی ارجاعی:** محدودیت کلید خارجی (Foreign Key) جداول فرزند را به جداول والد لینک می‌کند؛ به همین دلیل، [Referential Integrity](#) آن‌ها حفظ خواهد شد. علاوه‌براین، با تعریف کردن محدودیت Unique، از وجود داده‌های تکراری جلوگیری می‌شود. بنابراین، [افزونگی](#) (Redundancy) کمینه‌شده و هر رکورد، نمایان‌گر داده‌های منحصربه‌فردی خواهد بود.
 - **خودکارسازی اعتبارسنجی داده‌ها:** با تعریف Constraint در SQL Server ، داده‌ها به‌صورت خودکار در زمان درج و آپدیت، اعتبارسنجی خواهند شد و شما دیگر به بررسی Manual آن‌ها نیاز ندارید.
- موارد فوق، بر اهمیت تعریف محدودیت در SQL Server دلالت دارند. به‌طور کلی، وجود Constraint ها، مزیت‌هایی همچون بهبود کیفیت، سازگاری داده‌ها و همچنین، تسهیل مدیریت دیتا را به همراه دارد.

انواع Constraint در SQL Server

رایج‌ترین نوع های محدودیت در SQL Server به شرح زیر است:



۱- محدودیت NOT NULL

محدودیت NOT NULL در SQL Server ، از خالی یا اصطلاحاً Null بودن یک ستون خاص از جدول پایگاه داده جلوگیری می‌کند. به کمک این نوع از محدودیت، مطمئن می‌شوید که تمامی رکوردهای آن ستون خاص حاوی مقدار باشند. می‌توان محدودیت NOT NULL را به‌نوعی شبیه به شرایط کلید اصلی (Primary Key) در نظر داشت که مقدار آن نمی‌تواند Null باشد. توجه کنید که این دو، مفاهیم مختلفی هستند و برای ملموس‌تر شدن محدودیت‌ها از این مثال استفاده شد.

• ایجاد جدول جدید و اعمال محدودیت NOT NULL

معمولاً محدودیت در SQL Server به‌همراه عبارت Create Table مورد استفاده قرار می‌گیرد. به‌صورت پیش‌فرض، ستون‌های جدول می‌توانند مقدار Null داشته باشند. برای درک اهمیت محدودیت NOT NULL ، به مثال زیر توجه کنید:

در کوئری پایین، محدودیت NOT NULL به‌همراه ساخت یک جدول جدید مورد بررسی قرار می‌گیرد.

```
CREATE TABLE NewProducts (
    ProductID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    ProductName NVARCHAR(40) NOT NULL,
    SupplierID INT,
    UnitPrice DECIMAL(10,2),
    Discontinued BIT);
```

```
SQLQuery2.sql - D:\...URC70\Lenovo (56))*
CREATE TABLE NewProducts (
    ProductID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    ProductName NVARCHAR(40) NOT NULL,
    SupplierID INT,
    UnitPrice DECIMAL(10,2),
    Discontinued BIT
);
```

150 %

Messages

Commands completed successfully.

Completion time: 2024-02-03T12:06:34.2287987+03:30

با اجرای کوئری فوق، یک جدول تحت عنوان NewProducts ایجاد می‌شود. همانطور که در تصویر زیر قابل مشاهده است، دو ستون ProductID و ProductName نمی‌توانند مقدار Null داشته باشند. توجه کنید که در این مثال، ProductID به عنوان کلید اصلی جدول انتخاب شده است؛ بنابراین، مقدار آن به صورت ضمنی NOT NULL در نظر گرفته خواهد شد.

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
ProductName	nvarchar(40)	<input type="checkbox"/>
SupplierID	int	<input checked="" type="checkbox"/>
UnitPrice	decimal(10, 2)	<input checked="" type="checkbox"/>
Discontinued	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

• افزودن محدودیت NOT NULL به یک ستون موجود

شما می‌توانید با استفاده از ALTER TABLE و ALTER Column تغییرات موردنیاز روی جدول و ستون آن را لحاظ کنید. با اجرای کوئری فوق، این تضمین به وجود می‌آید که تمامی رکوردهای کارمندان، منحصر به فرد هستند و نمی‌توانند مقدار NULL بگیرند.

```
ALTER TABLE Employees
ALTER COLUMN EmployeeID INT NOT NULL;
```

توجه کنید که EmployeeID کلید اصلی این جدول است.

Column Name	Data Type	Allow Nulls
EmployeeID	int	<input type="checkbox"/>
LastName	nvarchar(20)	<input type="checkbox"/>
FirstName	nvarchar(10)	<input type="checkbox"/>
Title	nvarchar(30)	<input checked="" type="checkbox"/>
TitleOfCourtesy	nvarchar(25)	<input checked="" type="checkbox"/>
BirthDate	datetime	<input checked="" type="checkbox"/>
HireDate	datetime	<input checked="" type="checkbox"/>
Address	nvarchar(60)	<input checked="" type="checkbox"/>
City	nvarchar(15)	<input checked="" type="checkbox"/>
Region	nvarchar(15)	<input checked="" type="checkbox"/>
PostalCode	nvarchar(10)	<input checked="" type="checkbox"/>
Country	nvarchar(15)	<input checked="" type="checkbox"/>
HomePhone	nvarchar(24)	<input checked="" type="checkbox"/>
Extension	nvarchar(4)	<input checked="" type="checkbox"/>
Photo	image	<input checked="" type="checkbox"/>
Notes	ntext	<input checked="" type="checkbox"/>
ReportsTo	int	<input checked="" type="checkbox"/>
PhotoPath	nvarchar(255)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

• ایجاد محدودیت چندستونی با NOT NULL Constraint

شما می‌توانید با اعمال محدودیت در SQL Server، برای چند ستون جدول Constraint تعیین کنید. با اجرای کوئری زیر، یک جدول تحت عنوان OrdersArchive در پایگاه داده ساخته می‌شود. در این جدول، ستون‌ها نمی‌توانند مقدار Null بگیرند. ضمن اینکه یک «کلید اصلی ترکیبی» (PK_OrdersArchive) روی دو ستون این جدول تعریف شده است. این شرایط تضمین می‌کند که هر رکورد نشان‌دهنده یک سفارش یونیک است.

```
CREATE TABLE OrdersArchive (
    OrderID INT NOT NULL,
    CustomerID NCHAR(5) NOT NULL,
    EmployeeID INT NOT NULL,
    OrderDate DATETIME NOT NULL,
    CONSTRAINT PK_OrdersArchive PRIMARY KEY (OrderID, CustomerID)
);
```

همانطور که در تصویر زیر مشاهده می‌کنید، این جدول با شرایط مذکور در پایگاه داده ایجاد شده است.

Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
CustomerID	nchar(5)	<input type="checkbox"/>
EmployeeID	int	<input type="checkbox"/>
OrderDate	datetime	<input type="checkbox"/>

۲- محدودیت UNIQUE

اگر محدودیت Unique روی یک جدول دیتابیس اعمال شود، خیال شما از بابت منحصر به فرد بودن مقادیر یک یا چند ستون خاص راحت است. به بیان دیگر، امکان درج داده‌های تکراری یا اصطلاحاً Duplicated Data وجود ندارد. در مثال زیر، یک جدول به نام NewCustomers ایجاد خواهیم کرد؛ به طوری که CustomerID کلید اصلی و Email دارای محدودیت نوع Unique باشد.

```
CREATE TABLE NewCustomers (
    CustomerID INT PRIMARY KEY,
    Email NVARCHAR(50) UNIQUE
);
```

استفاده از این نوع محدودیت در SQL Server ، این تضمین را ایجاد می‌کند که دو مشتری یک آدرس ایمیل یکسان نداشته باشند.

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
Email	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

۳- محدودیت PRIMARY KEY

Primary Key Constraint نوعی محدودیت در SQL Server است که نشان می‌دهد هر رکورد از جدول یکتا است. به واسطه Primary Key Constraint، یکپارچگی دیتا و استخراج داده‌ها به صورت کارآمد انجام خواهند شد. توجه کنید که هر جدول، تنها امکان داشتن یک کلید اصلی دارد و این کلید نباید تکراری یا NULL باشد.

• افزودن محدودیت Primary Key در زمان ساخت جدول

```
CREATE TABLE CustomersTable (
    CustomerID INT PRIMARY KEY,
    CustomerName NVARCHAR(50),
    Email NVARCHAR(100)
);
```

با اجرای کوئری فوق، یک جدول به نام CustomersTable ایجاد می‌شود؛ در این جدول، CustomerID به عنوان کلید اصلی تعریف شده است. به این ترتیب، یک محدودیت Primary Key روی این جدول تعیین شد.

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
CustomerName	nvarchar(50)	<input checked="" type="checkbox"/>
Email	nvarchar(100)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

- اعمال محدودیت کلید اصلی روی یک جدول موجود

ALTER TABLE Orders

ADD CONSTRAINT PK_Orders **PRIMARY KEY** (OrderID);

شما می‌توانید این نوع محدودیت در SQL Server را روی یکی از جداول فعلی دیتابیس خود اعمال کنید. با اجرای کوئری بالا، روی ستون OrderID از جدول سفارش‌ها (Orders)، محدودیت PK لحاظ خواهد شد.

Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
CustomerID	nchar(5)	<input checked="" type="checkbox"/>
EmployeeID	int	<input checked="" type="checkbox"/>
OrderDate	datetime	<input checked="" type="checkbox"/>
RequiredDate	datetime	<input checked="" type="checkbox"/>
ShippedDate	datetime	<input checked="" type="checkbox"/>
ShipVia	int	<input checked="" type="checkbox"/>
Freight	money	<input checked="" type="checkbox"/>
ShipName	nvarchar(40)	<input checked="" type="checkbox"/>
ShipAddress	nvarchar(60)	<input checked="" type="checkbox"/>
ShipCity	nvarchar(15)	<input checked="" type="checkbox"/>
ShipRegion	nvarchar(15)	<input checked="" type="checkbox"/>
ShipPostalCode	nvarchar(10)	<input checked="" type="checkbox"/>
ShipCountry	nvarchar(15)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

۴- محدودیت FOREIGN KEY

محدودیت کلید خارجی (Foreign Key) نقش بسزایی در حفظ یکپارچگی ارجاعی میان جداول دیتابیس دارا است. به واسطه این نوع از محدودیت در SQL Server، ارتباط میان دو جدول، با ارجاع به کلید اصلی در یک جدول و یک یا چند ستون در جدول دیگر امکان پذیر خواهد شد.

اگر کوئری فوق را اجرا کنید، یک جدول به نام OrderTable ایجاد می شود که در آن OrderID به عنوان کلید اصلی، یکتا است. OrderID از طریق کلید خارجی CustomerID به جدول Customers متصل می شود. به این ترتیب، این اطمینان به وجود می آید که هر سفارش به یک مشتری حقیقی ارجاع داده خواهد شد.

```
CREATE TABLE OrderTable (
    OrderID INT PRIMARY KEY,
    CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID)
);
```

۵- محدودیت CHECK

محدودیت Check در SQL Server یک قانون محسوب می شود که مقادیر مجاز برای یک یا چند ستون جدول را تعیین می کند. این Constraint، سازگاری و دقت در دیتابیس را بهبود می بخشد. به کوئری زیر توجه کنید:

```
CREATE TABLE ProductsTable (
    ProductID INT PRIMARY KEY,
    UnitPrice DECIMAL(18, 2) CHECK (UnitPrice > 0),
    UnitsInStock INT CHECK (UnitsInStock >= 0)
);
```

پس از اجرای این Query، یک جدول به نام ProductsTable ایجاد می شود؛ قسمت آخر این کوئری، بررسی می شود که UnitPrice بزرگتر از صفر و UnitsInStock بزرگتر یا مساوی صفر باشد. این موضوع از طریق محدودیت Check اعمال شده است.

Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
UnitPrice	decimal(18, 2)	<input checked="" type="checkbox"/>
UnitsInStock	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

۶- محدودیت DEFAULT

فرض کنید محدودیت Default روی یک ستون از جدول دیتابیس اعمال شده و شما یک رکورد به آن اضافه کرده‌اید. در چنین شرایطی، یک مقدار پیش فرض برای آن ستون وجود خواهد داشت. با کمک این محدودیت در SQL Server، اطمینان خواهید داشت که مقادیر ستون مربوطه سازگار هستند و یکپارچگی داده‌ها حفظ می‌شود.

```
ALTER TABLE OrderDetails  
ADD CONSTRAINT DF_Quantity DEFAULT 1 FOR Quantity;
```

در مثال فوق، محدودیت Default به نام DF_Quantity برای جدول OrderDetails افزوده می‌شود؛ در این محدودیت، عدد یک به عنوان مقدار پیش فرض برای ستون Quantity در نظر گرفته شده است.

کلام آخر: کاربرد محدودیت ها در SQL Server چیست؟

محدودیت ها در SQL Server به شما کمک خواهند کرد تا انسجام، امنیت و تطبیق پذیری داده‌های خود در دیتابیس را بهبود دهید. در این مقاله، به بررسی انواع Constraint در SQL Server پرداختیم و شما با تعدادی از موارد استفاده آن‌ها آشنا شدید. حال با شناخت هرکدام، آیا می‌توانید مثال‌های دیگری از کاربرد انواع Constraint نام ببرید؟