











Microsoft SQL Server Kullanıcı Oluşturma ve Veri Tabanında Kolon Bazlı Yetki Verme

Merhaba arkadaşlar, veri tabanı yönetim sistemleri denince akla gelen hatta evrensel bir yapı olmasına rağmen sql denince akla ilk gelen Microsoft Sql Server üzerinde yeni kullanıcı oluşturma ve bu oluşturduğumuz kullanıcıya sadece bir veri tabanı içerisinde ki iki kolona select yapma yetkisi vereceğiz. Bu iki kolona select yapma dışında başka herhangi bir işlem yapamayacaktır. İşlemlerimize ekran görüntüleri ile başlayalım:

1. Sunucu seviyesinde “Logins” başlığını sağ tıklayıp “New Login...” seçeneği tıklanır

2. “New Login...” tıkladıktan sonra açılan kullanıcı oluşturma ekranında oluşturacağımız kullanıcı adı ve Sql Server Doğrulaması için şifreyi oluşturuyoruz.

3. “Server Roles” kısmında kullanıcının sunucu üzerinde sahip olacağı rolleri seçiyoruz. Bu yazımızda sınırlı yetkili kullanıcı oluşturduğumuzdan kullanıcıyı public rolünde bırakıyoruz.

4. “User Mapping” alanından kullanıcıya bağlanacak veri tabanlarını seçiyoruz. Biz sadece bir veri tabanını ve bu veri tabanına bağlı public rolünü seçerek devam ediyoruz.

5. Status alanından kullanıcı aktif ve giriş izinli olarak seçtikten sonra OK butonuna basarak kullanıcıyı oluşturuyoruz.


Buraya kadar sunucu üzerinde kullanıcı oluşturmaya gördük. Bu ayarlar ile oluşturulan kullanıcı her hangi bir işleme yetkili değildir. Yazımızın bundan sonraki kısmında oluşturduğumuz bu

kullanıcıya bağlanan veri tabanında Users tablosundan Sid ve UserSettingsXml kolonlarına select işlemi için yetki verilecek. Bu işlemler dışında kullanıcı işlem yapamayacak.

1. Kullanıcıya bağlanan veri tabanı içinde Security > Users altından oluşturduğumuz kullanıcıya çift tıklayarak veya sağ tıklayıp Properties seçeneğine tıklayarak kullanıcı ayarlarına ulaşıyoruz. Açılan kullanıcı ayarlarında Securables alanına geçiş yapıyoruz.

2. Securables alanında bulunan "Search..." butonuna tıklayarak yetki verilebilecek nesne türlerini listelerini bulup aradığımız TABLES seçeneğini bulacağımız ekranı açıyoruz.

3. Açılan "Add Objects" ekranında "Specified objects..." seçeneği seçili iken "OK" butonuna tıklayarak nesne türlerinin listesini barındıran ekranı açıyoruz.

4. Açılan ekrandan, bizim senaryomuzda sadece tablonun kolonları üzerinde yetki vereceğimiz için sadece "Tables" seçeneğini seçip "OK" butonuna tıklayarak veri tabanı içerisinde tanımlı olan tablo listesini barındıran ve seçim yapmamızı sağlayacak olan ekranı açıyoruz.

5. Açılan tablo listesinden gene baştaki senaryomuza uygun olarak sadece "Users" tablosunu seçip "OK" butonuna tıklayarak tablonun yetki listesine eklenmesini sağlıyoruz. Açılan ekrandan tekrar "OK" butonuna tıklayarak tablonun eklenme işlemini bitiriyoruz.

6. Bu hali ile kullanıcı hala public rolünde olduğundan işlem yapamayacaktır. Tabloyu yetki ekranına ekledikten sonra tablo listesinden select işlemi için Grant alanını seçiyoruz ve seçim işlemi select satırında iken "Column Permissions..." butonuna tıklayarak tabloda tanımlı sütun listesini ekrana getiriyoruz. Listedeki Sid ve UserSettingXml kolonlarına grant ile verip "OK" butonları ile işlemi bitiriyoruz.

Aşağıda işlem sonunda oluşturulan kullanıcı ile sunucuya bağlanılıp çalıştırılmaya çalışılan bazı sorgu ve sonuçlarını görebilirsiniz.



Tablo Seviyesinde Veri Bütünlüğü

Merhaba arkadaşlar,

Daha önceki yazımızda [sütun seviyesinde veri bütünlüğünü](#) incelemiştik. Bu yazımızda da tablo boyutunda veri bütünlüğünü inceleyeceğiz.

Tablo boyutunda veri bütünlüğünü temelde iki şekilde sağlayabiliriz.

Kolonlar Arası Constraint ile:

Bu versiyonda veri girişinde bir birine bağımlı olan kolonlar arasında CHECK constraint ayarlanarak veri bütünlüğü sağlanabilir. Örneğin: Kişiler tablosunda kayıt tarihi alanının doğum tarihi kolonundan büyük olma şartı gibi.

Foreign Key Ayarlama:

Tablonun belli bir alanına girilen değer başka bir tablonun tekil alanında olma şartı durumunda foreign key kullanılır. Örneğin: Kişiler tablosunda "Unvan_Kodu" alanı olsun. Bu alana girilen değer Unvanlar tablosunda olması gerekmektedir. Aksi

durumda kişinin unvanını gösteren koda karşılık bir unvan olmadığından tutarsızlık meydana gelecektir.

Foreign key olarak tanımlanan alan NULL değer taşıyabilir. Mesela kişiyi veri tabanına ilk kayıta unvanı tanımlanmamış olabilir, bu durumda unvan tanımlanana kadar alan NULL değer olarak atanacaktır.

- Foreign Key SQL Kullanarak Tanımlama
- SQL Studio Management ile Tanımlama:
Öncelikle referans alacak ve referans alınan tablolar database diyagrama eklenmeli, daha sonra referans alacak olan Kisiler.UnvanKod kolonu sürüklenerek Unvanlar tablosu üzerine bırakılır. Bu işlemden sonra aşağıdaki ekran açılacaktır. Bu ekranın sol tarafında Unvanlar tablosunun UnvanKod kolonundan referans alınacağını ve sağ tarafında da Kisiler tablosunun UnvanKod alanının referans alacağını belirtiliyor.



Foreign key tanımında referans alınan kolonda değişiklik veya silme olması durumunda referans alan kolonda hangi işlemin olması gerektiğini de belirtmemiz gerekmektedir aksi durumda bir veri tutarsızlığı durumu ortaya çıkar. Mesela daha önce “Müdür” unvanının kodu 1, “Şef” unvanının kodunun 2 olduğunu varsayalım. Herhangi bir sebepten dolayı “Müdür” unvanının kodu 2 olarak değiştirilmesi gerekirse ve bu değişiklik Kisiler tablosuna da işlenmezse daha önce şef pozisyonunda olan her bir kişi veri tabanında artık müdür olarak görünecek. Bunun önüne geçmek için SQL Server foreign key alanı üzerinde ON DELETE ve ON UPDATE işleminde hangi işlemin olacağını belirtebiliriz.

İfadenin genel tanımı aşağıdaki gibidir:

- ON DELETE: Referans alınan değer silindiğinde
- ON UPDATE: Referans alınan değer güncellendiğinde yapılacak işlemi ifade eder.

Bu ifadelerden sonra girilecek deęerler:

- CASCADE: Referans alınan deęer üzerinde yapılan deęişiklik referans alan deęerler üzerinde de işlenir. Referans alınan deęerin silinmesi durumunda referans alan kayıtlar da silinir.
- SET DEFAULT: Referans alınan deęerin silindięinde, referans alan kayıttta bir default tanım var ise deęer default olarak deęiştirilir.
- SET NULL: Referans alınan deęerin silindięinde, referans alan kayıt NULL deęer kabul ediyor ise deęer NULL olarak deęiştirilir.
- NO ACTION: Referans alınan deęer silindięinde, referans alan bir kayıt var ise silinmesine müsaade edilmez, varsayılan olarak atanan işlem budur.

Bu işlem SQL Managenet Studio üzerinden yapıldıęında aşıęıdaki ekranda görünen alandan yapılabilir.



Kolon Seviyesinde Veri Bütünlüęünü Saęlamak

Kolon seviyesinde tanımlanan şartlar (constraint) aynı tablo ięerisinde kolonda yer alan deęerlerin tutarlılıęını sorgular.

Kolon seviyesinde veri bütünlüęü saęlama yollarını inceleyelim;

Primary Key:

Normalizasyon kuralları geređi her tabloda en az bir anahtar alan bulunmak zorundadır. Tanımlanan bu anahtar alan primary key olarak tanımlanır ve NULL deđer kabul etmeyen tekil bir anahtardır. Tablo üzerinde clustered bir anahtar tanımlı deđilse ve primary key tanımında NONCLUSTERED deđeri kullanılmadıysa clustered yapıda oluşturulur. Her tabloda en fazla bir tane olabilir.

Primary key tanımlama için kullanılan genel ifade:

Var olan tablo üzerinde PRIMARY KEY tanımlama:

UNIQUE Constraint:

Primary Key tanımında sadece bir tane olabileceđini belirtmiřtir. Unique şartında ise birden fazla olabilir. Unique index tanımlandıđı kolonda her deđerden en fazla bir tane bulunabilir. NULL deđer taşıyan bir satır olması durumunda ikinci bir NULL geldiđinde kabul etmeyecektir. Eđer var ise tekil yok ise NULL deđer taşıyabilir diye bir şart olması durumunda bunu daha önceki **ŞARTLI INDEX** yazımızdan faydalanarak yapabilirsiniz.

Tablo Oluřturma Sırasında UNIQUE INDEX Oluřturma:

Tanımlı Tablo Üzerinde UNIQUE Tanımlama:

Şartlı INDEX (Filtered Index) Oluşturma

Merhabalar, bu yazımızda daha önceki çalışmalarında öğrendiğim, üzerinde unique index tanımlanan null değer alabilen kolonda ikinci bir null değer geldiğinde bu kaydın unique indexten dolayı kaydedilmeyeceği, bilgisinin yeni [sql server](#) versiyonlarında artık şartlı index yazarak geçersiz olabileceğini ve nasıl yapılacağını öğreneceğiz.

Örneğin bir kişinin kredi kartı numarası olmak zorunda değil (nullable) ama var ise kişiye özel bir numaradır yani unique.

Şartlı INDEX yazmak aslında normal index yazmaktan çok farkı yoktur. Yapılması gereken index ifadesinden sonra WHERE ile koşulların yazılmasıdır. Bu şekilde index sadece şartın sağlandığı durumlarda geçerli olur.

Bu şekilde sadece [Name] kolonu üzerinde sadece [Name] kolonunun null olmadığı zaman çalışacak bir unique index tanımlamış oluruz.

Sql İle Kimlik No Kontrol

Kişi kaydı bulunan bilgi sistemlerinin hemen hemen hepsinin en önemli alanı kimlik numarasıdır. Kimlik numaraları de rastgele oluşturulmuş bir sayı değil de belli bir algoritmaya göre

oluřturulmuř bir numaradır.

Ařaęıdaki sql fonksiyonun kullanarak sisteminize kaydedilen kayıtların uygun kayıtlar olup olmadığını sorgulayabilirsiniz. Bir fikir olarak mesela, tabloya yazılacak bir trigger ile bütün kayıtlar otomatik olarak sorgulanabilir.

Fonksiyonun kullanımına örnekler:

Bir kaydı kontrol etmek için örnek

Kayıt tablosunda kimlik numaraları hatalı olan kayıtları getirmek için de ařaęıdaki gibi bir sorgulama yapılabilir.

SQL Server Trigger İle Güvenli Alıř Veriř

Merhaba arkadaşlar, bu yazımızda sql server üzerinde trigger özelliğini kullanarak güvenli alıř veriř sistemini simile etmeye çalışacağız.

NOT: Mail gönderebilmek için Sql Server üzerinde "Management > Database Mail" özelliğinin ayarlanmış ve aktifleştirilmiş olması gerekmektedir.

Senaryomuz şu şekilde olacak:

Sistem ayarlarında izin verilen max güvenli alış veriř miktarı kaydı bulunacak. Diđer bir tabloda da sistemin řüpheli bulduđu ve engellediđi marketlerin listesi bulunacak.

Sistemin alış veriř tablosuna bir alış veriř kaydı geldiđinde sipariř tutarı izin verilen max. tutardan fazla ise veya alış veriř yapılan market řüpheli marketler listesinde ise kullanıcıya bir bilgi maili atılacak ve iřlem iptal edilecek.

Bunun için gerekli tablolar scripti ařađıdaki gibidir.

Gerekli kontrolleri yapacak olan trigger kodumuz.

SQL Tarih Formatlama

[Veri tabanı](#) programlama çalışmalarında karşıma çıkan ve zamanında epey zorlayan bir konu ile yeniden beraberiz. Başlıkta da kendini belli ediyor [Microsoft Sql Server](#)'da görüntüleme iřlemi sırasında tarih formatlama;

Ařađıdaki kod parçasında bazı formatlama kodları ve çıktılarını görebilirsiniz:

Ms Sql Server Tablo Boyutlarını Görme

Veri tabanı yöneticilerinin kontrol ettikleri konulardan biri de veri tabanı boyutlarındaki artışlardır. Aşırı artan bir veri tabanı boyutunda hangi tablonun buna sebep olduğunu bilmek sorunun çözümüne giden yolu da belirleyecektir.

Boyut artışları tablolardaki kayıt sayısının artışı ile hesaplanabilir ancak tablodaki kolon sayısındaki değişkenlikten dolayı doğru sonuç vermeyecektir. Aşağıdaki `sql` kodundan faydalanılarak veri tabanında yer alan tabloların KB ve MB cinsinden boyutları görülebilir.

Sistem Nesneleri Dışında Veri Tabanından Her Şeyi Silme

Bazı veri tabanı çalışmalarında veri tabanında yer alan bütün nesnelerin silinmesi gerekebiliyor. Burada akla hemen şu gelebilir: madem her şeyi siliyoruz, neden veri tabanını tamamen silip yeniden oluşturmuyoruz, tabi ki yetkiden dolayı yapamıyor olabiliriz, bu durumda tek tek silme ile uğraşmak yerine bunu bir kod parçası ile yapabiliriz.

Sql Server Auto Increment (Otomatik Artan) Kolon Ayarlama

Veri tabanı çalışmalarında kullanılan en önemli başlıklardan biri de otomatik artan değerdir.

Örnek olarak sıra girilen kaydın sıra numarası veya Id kolonunun değeri otomatik artan değerde ayarlanması işlem kolaylığı sağlayacaktır.

Şimdi Sql Server üzerinde otomatik artan kolon (IDENTITY) nasıl ayarlanır, beraberce inceleyelim.

Identity kolonu ayarlanırken göz önünde bulunması gereken en önemli unsur her tabloda en fazla bir identity kolonu olabileceğidir.



Otomatik Artan Kolon

Otomatik artan kolon tablo dizayn ekranında iki farklı şekilde yapılmaktadır.

- İlki kolonunun özelliklerinden “Identity Specification” özelliğini “Yes” olarak ayarlanır. Bu özelliğin altında yer alan iki özellik: “Identity Increment” artışın kaçar kaçar olacağını, “Identity Seed” ise ilk sayının değerini belirtir.
- İkinci şekilde de resmin sağ tarafında yer alan tablo özelliklerinden “Identity Column” seçeneğini seçmektir. Bu şekilde ayarlanması durumunda yukarıdaki iki değer varsayılan olarak 1 olarak ayarlanacaktır.

Kod ile belirtilmek istendiğinde de: