

Karkas SQL Server Veri Tabanı Standartları

Atilla Özgür

December 23, 2013

1 Kurulum Notları

Veritabanı kurulurken, SQL Server içinden Collation olarak Turkish_CI_AS seçilmelidir. ■
Turkish_CI_AS Sunucu coallationu olarak seçilmelidir. Nchar unicode olarak değerleri tuttuğu için 2 katı yer kaplamakta ve daha yavaş çalışmaktadır. Eğer veritabanında aynı anda iki dil bilgisi tutulmayacaksa, yani hem rusça hemde türkçe bilgi, yazi değerleri veri tabanında varchar yada char olarak tutulmalıdır.

2 Şema Kavramı

SQL Server schema kavramını en iyi şekilde uygulayan veri tabanları arasındadır. ■
Bu yüzden schema yapısının SQL server veritabanı tasarımı yapılırken kullanılması tavsiye edilir. Bu yapıda her ana modul için bir ana şema, bir tanım tablosu şeması açılması tavsiye edilmektedir. Referential Integrity korunması açısından bütün uygulama tek bir veritabanında çalışmalıdır.

Örneğin ihtiyaç duyduğunuz modullerin İnsan Kaynakları, Muhasebe ve Bordro olduğunu düşünelim. Bu yapıda aşağıdaki şemaların açılması uygun olacaktır.

- MUHASEBE
- TT_MUHASEBE
- INSAN_KAYNAKLARI
- TT_INSAN_KAYNAKLARI
- BORDRO
- TT_BORDRO
- ORTAK
- TT_ORTAK
- OZLUK_BILGILERI
- TT_OZLUK_BILGILERI
- FIRMALAR
- TT_FIRMALAR

Burada muhasebe, insan kaynakları ve bordro şemaları zaten ihtiyaç duyulan şemalardır. Bunlara ek olarak kişi bilgilerinin tutulması için ad, soyad, kimlik bilgileri etc ORTAK şeması önerilmektedir. TT_ORTAK şeması buradaki bilgiler için gerekli tanım tablolarını tutacaktır. Örneğin TT_ORTAK.CINSIYET ismindeki bir tanım tablosu cinsiyet bilgileri tutabilir. Burada Kisi için tutulacak bilgiler herkes için kesin tutulması gereken bilgilerdir. Örneğin ad soyad, tc kimlik no resim gibi. Ama eğer sadece yazılımı kullanan kurumdaki, kişilere ait (örneğin kurum sicil no, kurum eposta, kurum telefon) gibi bilgiler var ise bunların farklı şemada tutulması daha yararlı olacaktır.

Buraya FIRMALAR şeması muhasebe firmalarının fatura kesecekleri firma bilgileri için eklenmiştir. Firma, Kisi bilgileri bir çok modul tarafından ihtiyaç duyulan bilgiler oldukları için bunların ayrı şemalarda tutulmaları daha yararlı olacaktır.

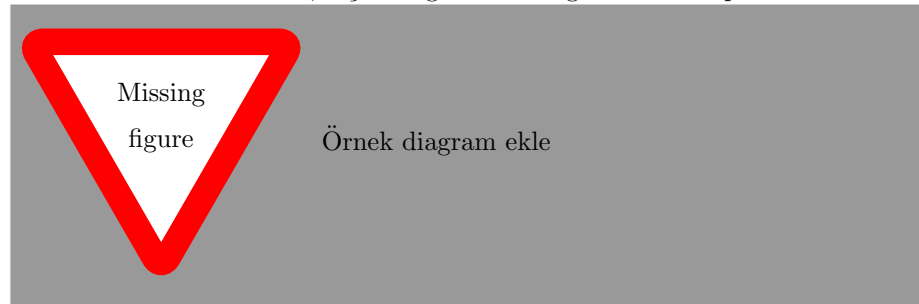
Son olarak Ozluk bilgileri, Bordro ve insan kaynaklarının ihtiyaç duydukları ortak bilgileri taşıyacaktır. Burada bu bilgiler sadece insan kaynakları şeması yada bordro şeması yerine ikisinin ortak kullandığı bir şemada olması daha uygundur. Yani diğer bir deyişle; eğer bir tablo birden fazla modul tarafından kullanılıyor ise, bu ortaklığı belirtecek yeni bir şema gerekirse açılmalıdır. Bu şema ismi eğer dışarıdan veri alınıyorsa bunu anlatan bir isim, eğer 2 modülün ortak noktası ise, bu ortak nokta olabilir. Sadece 2 modul tarafından kullanılacak bir tablo ORTAK şemasını eklenmemelidir.

Ana modul şemalarından 10 taneden fazla tablo bulunması bu modülün büyük olduğunu göstermektedir. Bu modul 2-3 ana modul olarak ayrılabilir. Örneğin personel için aşağıdaki gibi bir ayırım olabilir.

- PERSONEL
- PERSONEL_IZIN
- PERSONEL_HARCAMALAR

Ayrıca gecici daha sonra silinebilecek tabloların tutulması için GECICI veya TEMP isminde bir şemada eklenmesi uygun olacaktır.

Her modul relationları, ilişkileri gösteren diagramlara sahip olmalıdır.



2.1 Şema, Tablo, Kolon İsimlendirme

Şema ismi olarak kullandığımız isimleri tablo isimleri olarak kullanmayın. Yani eğer BORDRO diye bir şemanız varsa, bu şema içinde veya diğer şemaların içinde BORDRO diye bir tablonuz olmamalıdır.

VS.NET içinde proje ismi olarak kullanacağınız bir ismi şema ismi olarak kullanmayınız. Yada şema isimlerinizi, VS.NET içinde proje ismi olarak vermeyin. Derleyici (Compiler) için sorunlar çıkarabiliyor.

Veritabanındaki Tablo ve Şema isimleri tüm harfler büyük, Kelimeler arasında ise _ olacak şekilde seçilmelidir.

- INSAN_KAYNAKLARI.ILAC_ALIMI.ALINAN_ILACLAR
- ORTAK.KISI_EK_BILGILER
- MUHASEBE.FIS
- MUHASEBE.FIS_DETAY

2.1.1 Kolon İsimlendirme

Kolon isimleri verilirken .NET isimlendir konvansiyonuna uygun olarak her kelime büyük harf ile başlamalı, diğer harfler küçük harf olmalıdır. Kolon isimleri kullanılırken türkçe karakterler (İ,Ü,ı ..) kullanılmamalıdır. Bakınız aşağıdaki örnekler.

1. Ad
2. Soyad
3. IkinciAdi
4. TcKimlikNo
5. SehirNo

Raporlama açısından bütün tablo'ların ve anlamı açık olmayan kolanların descriptionlarının yazılması tavsiye edilir. Bu sayede veritabanından otomatik rapor çıkaran araçların raporları daha anlamlı olacaktır.

2.1.2 Kolon İsimlendirme - Primary Key

Primary key - Birincil anahtar isimlerinin nasıl olması gerektiğine proje başında karar verilmeli ve bu isimler proje boyunca uyulmalıdır. Buna göre birincil anahtar için aşağıdaki kelimelerden biri seçilmelidir.

1. Key
2. ID
3. Id

Key anahtar, Id Identity (kimlik) kelimelerinin kısaltmasıdır. Projede bunlardan sadece birinin kullanılması daha uygun olacaktır. Aynı şekilde ID ve Id olarak karar verilmeli ve hep aynı şekilde kullanılmalıdır.

Aşağıdaki örneklerde Key kelimesinin seçilmiştir.

Birincil Anahtarları TABLO ADI + Key olarak verilmesi tavsiye edilmektedir. Örneğin

Tablo Adı	Primary Key Adı
ORTAK.KISI	KisiKey
MUHASEBE.FIS	FisKey
BORDRO.ILAC_ALIMI_ALINAN_ILACLAR	IlacAlimiAlinanIlaclarKey
ORTAK.KISILEK_BILGILER	KisiEkBilgilerKey

Eğer tanım tablolarının primary key'ı seçiliyor ise Tabloİsmi + (No,Turu,TurNo,Tipi,TipNo) gibi kolon isminden sonra türkçe okunmaya uygun bir ek kullanılmalıdır.

Örneğin

Tablo Adı	Primary Key Adı
TT_ORTAK.Cinsiyet	CinsiyetTipNo
TT_ORTAK.SEHIR	SehirNo
TT_OZGECMIS.EGITIM_TIP	EgitimTipNo
TT_OZGECMIS.YABANCI_DIL_SINAV_TUR	YabancıDilSinavTurNo

2.1.3 Kolon İsimlendirme - Foreign Key

Kolonlarda ikincil anahtar isimleri verilirken dikkat edilmesi gereken kural, referans edilen tablonun bir tanım tablosunu yoksa bir ana tablosu olduğu kuralıdır. Eğer bir ana tabloya referans veriliyorsa Tabloİsmi + Key kullanılmalıdır. Eğer bir tanım tablosuna referans veriliyorsa Tabloİsmi + (No,Turu,TurNo,Tipi,TipNo) gibi kolon isminden sonra türkçe okunmaya uygun bir ek kullanılmalıdır. Bu sayede bir tablo incelenirken hangi tablolara referans verdiği daha rahat bir şekilde anlaşılacaktır. Ayrıca verilen referansın bir tanım tablosunamı yoksa bir ana tabloyamı yapıldığı daha iyi anlaşılacaktır.

Not eğer verilen ikincil anahtar kolonu bir iş gereği yüzünden ise, kolon isminde bu durumun belirtilmesi ve aynı zamanda tablo isminin kullanılması daha iyi olur.

Örneğin Bilgi Edinme için tasarlanan tablo isimlerine bir bakalım.

- BILGLEDINME.DILEKCE
 - DilekceKey
 - BasvuruSahibiKisiKey
 - GonderenIP
 - DilekceIcerik
 - CevapIstekTip
 - GelisYoluTip
 - VatandasBasvuruReferansKey

Bu tabloyu okuduğumuzda DilekceKey primary key, BasvuruSahibiKisiKey ve VatandasBasvuruReferansKey ana tablolara referans eden foreign key, CevapIstekTip ve GelisYoluTip tanım tablolarına referans eden bilgiler.

Burada BasvuruSahibiKisiKey ORTAK.KISI tablosuna büyük ihtimal ile referans vermektedir. KisiKey yerine BasvuruSahibiKisiKey kullanılması bu anahtar dilekce başvuru ifade ettiğini göstermektedir. Bu anahtar kullanılarak kişi adı soyadı gibi bilgilere ulaşılabilir.

CevapIstekTip ve GelisYoluTip kolonları ile TT_BILGLEDINME şemasında CEVAP_ISTEK ve GELIS_YOLU isminde tablolar olduğunu biliyoruz.

2.2 Kolon Veri Yapıları

2.2.1 Primary Key

Birincil anahtar (Primary Key) konulmayan tablo olmayacaktır. Primary key değerleri olarak uniqueidentifier veya Identity seçilmeli proje boyunca benzer bir yapı kullanılmalıdır. Primary key Sentetik Anahtar (Surrogate Key) olarak int IDENTITY değerleri yerine uniqueidentifier tercih edilmesi, kod yazılması kolaylığı açısından değerlendirilmeli ve proje başlangıcında karar verilmelidir. Uniqueidentifier eğer PK olarak kullanılıyorsa default değer olarak newid() olmalıdır.

2.2.2 Foreign Key

Eğer bir tablo içindeki kolon başka bir tabloya referans veriyorsa kesinlikle foreign key tanımlanmalı ve bu kolonun ismi TabloİsmiTipNo yada TabloİsmiKey olarak tanımlanmalı. TipNo tanım tabloları için. Key ise normal tablolar için kullanılmalıdır. TipNo yerine referans verilen tanımın türkçesine göre TurNo gibi diğer kelimelerde kullanılabilir. Eğer bir kolon ismi TurNo,TipNo,Turu,Tipi,RenkNo,BilgiNo gibi bir kelime ile bitiyorsa ise çok büyük ihtimal ile Tanım Tablolarına referans veriyordur. TipNo(...) okunduğu zaman bir tanım tablosuna referans düşünülürken, Key ise 1..* veya 1..1 ilişkisini tanımlamalıdır.

2.2.3 Yazı Değerler

Yazi degerleri eger deęişken ise varchar ile, eęer sabit bir deęer ise char olarak tutulmalıdır. Örneęin:

- Eęer bilgi girilen tip daima aynı sayıda ise örnek: Posta Kodu gibi char olarak tanımlanmalıdır.
- Eęer bilgi girilen deęer deęişken bir deęer ise, ad soyad gibi varchar olarak tanımlanmalıdır.

Eęer veri tabanında kullanılan genel dilden (türkçe olmayan) farklı deęer girilebilir ise nchar,nvarchar tanımlanmalıdır.

2.2.4 Sayı Değerler

Sayı deęerleri için alabilecekleri aralıkları düşünerek tiny int, short int , int veya big int kullanınız.

- Tablodaki deęer -32,768 - 32,767 smallint veri yapısını kullanın.
- eęer tamsayı deęer -2,147,483,648 - 2,147,483,647 arası ise int veri yapısını kullanın.

2.2.5 Para ve Ondalık Değerler

- Money SQL standartında olmadığından kullanılmamalıdır.
- Float ve double tam deęerler deęildir, yaklaşık deęer olarak çalışırlar. Para hesaplarında kesinlikle kullanılmamaları gerekir.

- Para değerleri numeric(18,2) veya decimal(18,2) olmalıdır. numeric ve decimal bir birlerinin alias'ıdır.
- Eğer para değerlerinde daha fazla küsürata ihtiyaç var ise numeric(18,4)-decimal(18,4) gibi artırılabilir.

2.2.6 Tarih Değerler

Tarih değerleri datetime (3.33 milliseconds accuracy) Jan 1, 1753 — Dec 31, 9999 ,smalldatetime (1 minute accuracy) Jan 1, 1900 — Jun 6, 2079 değerleri arasını göstermektedir. .NET üstünde doğru kabul edilen, örn: 01.01.1600 gibi bir değer SQL Server üzerinde hata verebilir. Tarih değerlerinin bu aralıkta olduğuna emin olunuz.

- Tarih bilgilerini tutarken smalldatetime tercih ediniz.
- Eğer sadece yıl bilgisi tutacaksanız, Mezuniyet yılı smallint tutabilirsiniz. Bu tür bir kolona constraint ekleyin.örn: `CHECK([GirisSene] >= (1900)AND[GirisSene] <= datepart(year, getdate()))` Yukarıdaki kısıtlama giriş senesinin 1900 ile şu an bulunduğumuz sene arasında kalmasını sağlamaktadır.

2.2.7 Resim Dosya gibi Binary Değerler

Resim veya binary data tutmak için varbinary(max) tercih edin. image veri tipi SQL Server'ın yeni versiyonlarında kullanılmayacaktır.

2.2.8 Diğer Notlar

- Tablolarda mümkün olduğunca girilen bilgiler not null yapılmalıdır.
- Eğer anlamlı bir default/varsayılan değer verilebiliyorsa verilmelidir. Örneğin GuncellemeTarih default now() verilebilir.

3 Yazılım Geliştirme Yetkileri

Yazılım geliştiriciler için sınırlı yetkili veri tabanı kullanıcılar açılması ve SQL Sever Management Studio üzerinde işlem yapılırken bu kullanıcılar ile sisteme girilmesi düşünülmelidir. En doğrusu yazılım geliştirici kullanıcılarının bir role atanması ve bu rolün kısıtlanması olabilir.

4 Stored Procedures - Functions

Yazılım geliştiriciler Stored Procedure yazma gereksinimi duyarlarsa bunu veri-tabanı yöneticisi ve takım liderlerinin onayını aldıktan sonra yapabileceklerdir. Veri tabanında mümkün olduğunca stored procedure ve trigger olmamalı, bunların yerine .NET kodları ile çözüm bulunmaya çalışılmalıdır.

Stored Procudures usp_ ile, functionlar ufn_ ile, triggerler ise utnX_ ile başlamalıdır (X-i;insert, u:update, d:delete). Note: eğer bir stored Procedure sp veya xp ile başlarsa SQL server bunun System stored procedure'u olduğunu düşündüğü için ilk önce master veritabanında arayacaktır.

5 İsimlendirme Konvansiyonu

5.1 İsimlendirme Konvansiyonu - Stored Procedure - Gömülü Yordamlar

Stored Procedure'lara isim verirken tablo ismi + yaptığı işlem şeklinde isim verin.

- Tablo Ismi + yaptığı iş
- usp_PersonelAraAdiVeSoyadiIle
- usp_Personel_Oku_Tum
- usp_MuhasebeYillikMizanHesapla
- usp_BordroAylıkBordroHesapla

Not bu isimler örnek olsun diye verilmiştir. Veritabanında basit CRUD ve arama işlemleri için stored procedure olmamalıdır.

5.2 İsimlendirme Konvansiyonu - Table and View-Tablo ve Görüntü

- INSAN_KAYNAKLARI.MUSTERI
- TANIM_TABLOLARI.NUFUS_CUZDANI_VERILIS_NEDENI

Schema ismi + Tablo ismi, 2 kelimedenden fazla olan isimler _ ile ayrılıyor. Tablo ve görüntülere (view) ayrı isimler vermiyoruz.

5.3 İsimlendirme Konvansiyonu - Column-Kolon

Her kelimenin ilk harfi büyük yazılır.

- MusteriKey
- SonDegistirmeTarihi
- TipNo
- MusteriKey

5.4 İsimlendirme Konvansiyonu - Index

- IX_TabloIsmiKolonIsmi
- IX_MusteriMusteriNo
- TipNo
- MusteriKey

Her kelimenin ilk harfi büyük yazılır.

5.5 İsimlendirme Konvansiyonu - Yabancı anahtar (Foreign key)

- FK_TabloIsmi1KolonIsmi1FK_TabloIsmi2KolonIsmi2
- FK_MusteriMusteriNoSiparisMusteriNo

Her kelimenin ilk harfi büyük yazılır.

5.6 İsimlendirme Konvansiyonu - Default - Varsayılan

- DF_
- DF_PERSONEL_IlkGirisTarihi

SQL Server DF veriyor. Management Studio icinde Default'ları verirsiniz, İsimlendirme düzgün olur.

5.7 İsimlendirme Konvansiyonu - Constraint

- CK_
- CK_OKUL_BILGILERI_GirisSenesi

SQL Server CK veriyor. Management Studio icinde Constarint yazarsanız, İsimlendirme düzgün olur.

5.8 İsimlendirme Konvansiyonu - Trigger - Tetikleyici

utrX + Tablo Ismi, yerine triggerin hangi işlem için olduğunu söyleyen bir kelime gelecektir.

- utrX_ + tablo ismi + işlem (insert (i), update (u), delete (d))
- utrX_ORTAK_KISI_Delete

Revision	Yazarlar	Tarih	Yapılan Değişiklikler
0.1	Atilla Özgür	2006	İlk Yazım
0.2	Atilla Özgür	2008	Fatih ile İç İşlerinde yapılan değişiklik
0.3	Atilla Özgür	2013-02-13	Çok uzun bir aradan sonra bazı düzeltmeler
0.4	Atilla Özgür	2013-07-04	Latex cevrimi, sectionlara ayırma , tümüyle tekrar yazıldı.