**VERİ TABANI - 1**

Bahar Dönemi

1. **VERİ TABANI KAVRAMLARI**
   1. **TABLO**

Günümüzde kullanılan veri tabanlarının birçoğu verileri tablolarda saklar. Tablolar, verilerin satırlar ve sütunlardan halinde düzenlenmesiyle oluşan veri grubudur.

Tablo içindeki her bir bilgi kayıt (satır) olarak isimlendirilir. Sütunlar ise alan olarak isimlendirilir.

Üniversitemize kayıtlı olan öğrencilerin kişisel bilgilerinin tutulduğu tabloyu düşünelim. Tablodaki her satır bir öğrenciyi temsil eder. Her sütun öğrencinin bir özelliğini temsil eder ve bu sütunlara isim verilir. Öğrenci adı, öğrenci soyadı, öğrenci numarası, doğum tarihi, telefon, adres vb.

Veri tabanlarında kullanılacak alan adlarının evrensel kullanıma uyması, veri tabanının başka bir yere kopyalanırken sorunsuz çalışması ve her ortamda kullanılabilmesi için standart isimlendirme yöntemleri kullanılır. Sütunlara verilen isimlerin küçük harflerle ve Türkçe karakter kullanmadan oluşturulması gerekir. Bazı veri tabanları Türkçe karaktere izin verse de hepsi desteklemediği için bundan kaçınılması gerekir.

Kelimeler arasında alt tirenin kullanılması tercih edilir. Tablolar içinden veri çağırırken nokta işareti kullanıldığı için kelimeler arasındaki boşluk yerine nokta koyulmamalıdır.

Tablo ismi belirlenirken, okunabilirliği artırmak için tablo olduğunun ve neye ait bir tablo olduğunun belirtilmesi gerekir. Güvenliği artırmak ve veri tabanına gelebilecek siber saldırıların erişimini azaltmak için tablo içeriği olduğu gibi tabloya isim olarak verilmemelidir.

Tablo içinde tutulan verilen olduğu gibi saklanacağı için birçok veri tabanı yönetim sisteminde Türkçe karakterler kullanarak satırlar oluşturulabilir.

Bir veri tabanı içerisinde birden fazla tablo kullanılabilir. İletişim tablosu, ders tablosu, bölümler tablosu gibi.

* 1. **VERİ TİPLERİ**

Oluşturulan veri tabanında tutulan verilerin hepsi aynı türde değildir. Tutulacak veriye göre değişiklik gösterir. Bu tabloda ne tür verilerin saklandığı veri türlerinden anlaşılır.

Ad – soyad karakterlerden oluşur. Doğum tarihi Date tipindedir. Ücretler sayı tipindedir. Hangi tip veri saklanacaksa o tercih edilerek belirtilir. Öğrenci numarası gibi matematiksel işlem yapılmayacak numaralar veri tabanında karakter olarak saklanabilir.

Her veri tabanının veri tipi tanımlaması farklıdır. Veri tabanında gereğinden fazla yer ayırmadan en uygun boyutta veri tipinin seçilmesi gerekir.

* + 1. **MSSQL Server**

**char(uzunluk):** Sayısal olmayan karakter türü verileri tanımlamak için kullanılır. Sütunda saklanacak veri boyutu ‘uzunluk’ parametresi ile belirlenir. ogr\_adi sütunu için char(20) yazılırsa en fazla 20 karakterlik bir metin yazılabileceği anlamına gelir.

**nchar(uzunluk):** char veri tipine benzer. Verileri ANSI formatında değil, Unicode formatında saklar.

**varchar(uzunluk):** char gibi karakter tipinde alanlar tanımlamak için kullanılır. Char tipinden farkı, karakter tipinin uzunluğunun değişebilmesidir. Parametre olarak verilen uzunluk maksimum uzunluktur. Veri daha kısa ise, uzunluğun gerektiği kadar alan kullanılır. 0-255 boyutunda veri saklayabilir.

**nvarchar(uzunluk):** varchar veri tipinin Unicode olarak veri saklayan halidir.

**text:** Çok uzun açıklamaları saklamak için kullanılan veri türüdür. 2 GB’a kadar veri girilebilen alfasayısal alanlar tanımlar.

**image**: Resim veri türüdür.

**int(sayı):** Tam sayı verileri saklamak için kullanılan veri türüdür. Bu veri türünde tanımlanan bir alana -2.147.483.648 ile +2.147.483.647 arasındaki değerler girilebilir.

**bigint(sayı):** Büyük sayısal alanlar tanımlamak için kullanılır. -9.223.372.036.854.777.808 ile +9.223.372.036.854.777.807 arasında değerler girilebilir.

**smallint:** Küçük tam sayısal alanlar tanımlamak için kullanılan veri türüdür. -32.768 ile +32.767 arasında değerler girilebilir.

**tinyint:** 0-255 arasında sayısal değerler alabilen veri tipidir.

**decimal, numeric:** Ondalıklı, kayan noktalı, virgüllü sayıların saklanması için gerekli alanları tanımlar.

**float**: Çok büyük ondalıklı sayıları saklayabilen veri türüdür.

**real**: Floattan küçük sayıları saklar.

**money, smallmmoney:** Parasal değerleri saklayan veri türüdür.

**datetime, smalldatetime:** Tarih, saat bilgilerini saklayan alanları tanımlamada kullanılır.

**binary**: İkili verileri saklayan alanları tanımlar.

**bit**: 0 veya 1 gibi sadece iki seçenekli verileri saklamak için gerekli alanları tanımlar.

**Neden Farklı Sayı Türleri Kullanılır?**

Veri tabanında yüzlerce, binlerce veri saklanacağını düşündüğümüzde hafızadan gereksiz büyüklükte kullanılmayacak yer ayırmak sistem maliyetini artırır. 1 bytelık veri tutacak iken 8 bytelık alanı boşa harcamamamız gerekir.

Tinyint: 1 byte veri saklar.

İnt: 4 byte veri saklar.

Örneğin, “yaş” alanına değer girileceği zaman 255’ten büyük bir sayının o alana veri olarak girilmesi imkansızdır. Bu alan için mssql server kullanırken int yerine tinyint kullanılması daha uygun olur.

* + 1. **Oracle**

**char(uzunluk):** 0-255 uzunluk arasında karakter tipi verileri saklamak için kullanılır.

**varchar(uzunluk):** Ayrılan alandan ihtiyacı olan kadarını kullanan veri türüdür.

**date:** Tarih, saat bilgilerini saklayan alanlar tanımlamak için kullanılır.

**integer**: Sayısal değerleri saklayan alanları tanımlar.

**number(tam, ondalık):** Ondalıklı sayısal değerler saklamak için kullanılır.

number(5,2); //5 basamak tam kısım, 2 basamak ondalıklı kısım için ayrılmış demektir.

**long:** Uzun sayısal alanlar için kullanılır.

**long row:** uzun ikili veri girilebilen alanlar tanımlamak için kullanılır.

**boolean:** Mantıksal iki seçenekli (true/false) verileri saklar.

* + 1. **MySQL**

**char():** 0-255 arasında karakter tanımlar.

**varchar():** 0-255 arasında ayrılan alandan ihtiyacı kadarının kullanılmasını sağlar.

**tinytext:** En fazla 255 karakterlik metinlerin yazılmasını sağlar.

**text**: En fazla 65.535 karakterlik metinlerin yazılmasını sağlar.

**tinyint:** -128 ile +127 arasında sayısal değerlerin tutulmasını sağlar. 1 bytelık veri saklar.

**smallint:** -32768 ile + 32767 arasında sayısal değerlerin tutulmasını sağlar. 2 bytelık veri saklar.

**mediumint:** -8388608 ile + 8388607 arasında sayısal değerlerin tutulmasını sağlar. 3 bytelık veri saklar.

**int**: -2147483648 ile 2147483647 arasında sayısal değerlerin tutulmasını sağlar. 4 bytelık veri saklar.

**bigint**: -263 ile +263-1 arasında sayısal değerlerin tutulmasını sağlar. 8 bytelık veri saklar.

**float(*tam*,ondalık), real(*tam*,ondalık) ve double precısıon(tam, ondalık):** Ondalıklı sayıları veri tabanında tutmak için kullanılır.

**date**: YYYY-MM-DD tipinde tarih verisini saklar.

**datetime**: YYYY-MM-DD HH:MM:SS tipinde tarih ve saat verisini saklar.

**timestamp**: YYYYMMDDHHMMSS tipinde tarih ve saat verisini saklar.

**time**: HH:MM:SS tipinde saat verisini saklar.

* 1. **KISITLAMALAR**

Tablo alanlarına veri girişini kontrol etmeyi amaçlayan düzenlemelerdir. Tabloları ilişkilendirmede, kullanıcı hatalarını önlemede, tekrarı önlemede vb. açısından önemlidir.

Sınav değerlendirme notlarının 0 ile 100 arasında bir değer girilmesi gerekir. Kısıtlar belirlenerek yanlışlıkla 200 gibi bir sayının sisteme girilmesinin önüne geçilmiş olur.

* + 1. **Primary Key (Birincil Anahtar)**

Tabloda tutulan verilerden her kaydın benzersiz olduğunu tanımlamak için kullanılan kısıtlamadır. Bir primary key değeri sadece bir veride bulunur.

TC kimlik numaranız sizin benzersiz bir birey olduğunuzu tanımlar ve bir kişiye özeldir. Tekrar ve başka birinde kullanılamaz. Öğrenci numaranız okul içerisinde 1 kişiye aittir, başka bir öğrenci o numara ile tanımlanamaz. Bu numaralar bizim için primary key olarak ifade edilir.

Primary Key olarak belirtilen bir alana veri girilmesi zorunludur. Boş bırakılamaz, değersiz (NULL) olarak veri kaydedilemez.

Bir tabloda bir kere birincil anahtar tanımlanabilir. Birden fazla sütun, birlikte birincil anahtar olarak tanımlanabilir. Bir tabloda birden fazla sütun ayrı ayrı birincil anahtar olarak tanımlanamaz.

* + 1. **Unique (Tekil)**

Tanımlandığı alandaki verilerin tekil, benzersiz olmasını sağlar. Primary Key’e benzer. Bu alana veri girilirken NULL olarak veri girilmesine izin verir. Birden fazla sütun unique olarak tanımlanabilir. Tablolarda tekrar eden verilerin girilmesini önlemek için kullanılır.

* + 1. **Foreign Key(Yabancı / İkincil Anahtar)**

Tablolar arası ilişkileri tanımlayan anahtardır. Bir tabloda birincil anahtar olarak tanımlanan alan, başka bir tabloda da kullanılacaksa, ortak kullanılan alanların ilişkilendirilmesi gerekir. Diğer tabloda Foreign Key olarak tanımlanır.

Tablolar arası bilgi tutarlılığının sağlanması, tabloların birbirine bağlanması için gereklidir.

* + 1. **DEFAULT**

Tanımlandığı alana bir değerin girilmemesi durumunda o alana, varsayılan olarak atanan değerin kaydedilmesini sağlar. Alana değer girilmediğinde NULL olarak kaydedilmesinin önüne geçilmiş olunur.

* + 1. **NOT NULL**

Tanımlandığı alana veri girişinin zorunlu olduğunu belirtir. Boş, değersiz, NULL olarak veri alanı doldurulmadan kayıt yapılmasını engeller.

**Kaynakça:**

* Özdemir S., “C# .Net 2008 ve ASP.NET”, Nirvana Yayınları, Ankara, 2008.
* Obalı M., “Oracle 10g”, Pusula Yayıncılık, İstanbul, 2007.
* Kaya Y. ve Tekin R., “Veritabanı ve Uygulamaları”, Papatya Yayıncılık, İstanbul, 2007.

Öğr. Gör. Kevser ONUR