

3. VERİ TABANI TEMEL KAVRAMLARI

Veritabanı (DataBase) : En genel tanımıyla, kullanım amacına uygun olarak düzenlenmiş veriler topluluğudur. Müşteri adres defterleri, ürün satış bilgilerinin saklandığı dosyalar, öğrenciler ve öğrenciler ait harç ve not bilgileri gibi, personel bilgi dosyaları gibi bilgi düzenleri veritabanlarına örnek olarak verilebilir. Belirli bir konu hakkında toplanmış veriler; bir veritabanı programı altında toplanırlar. İstenildiğinde toplanan bilgilerin tümü veya istenilen özelliklere uyanları görüntülenebilir, yazdırılabilir hatta bilgilerinden yeni bilgiler üretilerek bunlar çeşitli amaçlarla kullanılabilirler.

Veriler fiziksel hafızada Veri Dosyaları (DataFiles) halinde saklanırlar. Dosya, bilgisayarların bilgileri birbirinden ayırarak saklamak için kullandığı temel bilgi depolama yapısıdır. Bir dosyada, bir çok veri yer alabilir. Bir personel otomasyonu ele alınacak olursa, personel ile ilgili bilgiler, personelin çalıştığı birimler, meslekleri, aldığı maaş ile ilgili bilgiler aynı veri dosyasında ama farklı tablolar içerisinde yer alabilirler.

Bu bölümde veri tabanı ile ilgili temel kavramlar üzerinde durulacaktır. Öncelikle bu kavramlar tanımlanacak ve hangi amaçla kullanıldığı anlatılacaktır. Bu kavramlarla ilgili detaylı örnek ve açıklamalar sonraki bölümlerde yapılacaktır.

3.1. Tablo ve Elemanları

Tablo verilerin satırlar (row) ve sütunlar (column) halinde düzenlenmesiyle oluşan veri grubudur. Veritabanları bir veya daha fazla tablodan oluşurlar. Tablolar arasında ilişkiler düzenlenebilir. Tablonun satırlarındaki her bir bilgi kayıt (record), sütunlar ise alan (field) olarak isimlendirilir. Bir tabloda yer alan her bir kayıt bir satıra karşılık gelir. Örneğin personel listesi (yani personel tablosunu) ele alınacak olursa, her bir satırda bir personele ait bilgiler yer alır. Sütunlardaki alanlar (Field) ise yapılandırılmış bilginin her bir kısmını saklamak üzere yapılan tanımlamadır. Bir personele ait bilgilerin her biri sütunlarda tutulur. Personelin sicil numarası, adı, soyadı, çalıştığı birim, doğum tarihi gibi bilgilerin her biri bir sütun alanıdır. Her bir alan, yapılandırılmış verinin bir birimini tutmak üzere tanımlanır. Her bir sütunun adı ile birlikte diğer bilgilerinin (en fazla kaç birimlik bilgi bu hücrede saklanabilecek, ne tür bilgi saklanacak vs.) ortaya koyduğu tanıma alan denir.

Herhangi bir veritabanı programında çalışmaya başlamadan önce yapılacak işe uygun veri tabanı tasarımı yapılmalıdır. Bu işin en önemli aşamasıdır. Başlangıçta iyi tasarlanmayan bir veritabanı ileride geriye dönüşü olmayan verimsiz bir bilgi yığınına dönüşebilir. En basit hali ile veritabanı tasarımı; hangi tabloların olacağı, bu tablolarda hangi alanların olacağı, tablolar arasındaki alan ilişkilerinin neler olacağı ve alanlara ait özelliklerin tanımlanması yapılır. Alan özelliklerinde alan adı, alan tipi, alanın uzunluğu, alanın varsayılan değeri, bu alana yazılacak verilerin geçerlilik koşulları başlangıçta tasarlanması gerekir.

Veritabanının en önemli bileşeni tablodur. Her veritabanında en az bir tablo bulunur. Veritabanı işlemlerinde önce tablo/tablolara tanımlanır. Daha sonra tablolara kaydedilecek bilgilerin neler olacağı ve bu bilgilere ait özellikler tanımlanır. Personelin sicil numarası ve bunun sayılardan oluşması, personelin adı soyadı ve bunun harflerden oluşması gibi. Tanımlamalar bittikten sonra tablodaki bu alanlara ait gerçek bilgiler yazılır. Yazılan bu bilgiler tablolarda tutulur. Kayıt ile satır arasındaki temel fark, kayıt ile kastedilen yapının sütunlar hakkındaki bilgileri de içermesidir.

Tablolara girilmiş bilgilerden belirli şartlara uyanların liste şeklinde alınmasına **sorgu** adı verilir. Tablolardan gerektiğinde sorgulamalar yapılabilir. Değişik amaçlara göre sorgular hazırlanarak tablodaki bilgilerin tümü, bir kısmı veya belirli şartı sağlayanların listesi alınabilir. Örneğin, muhasebe bölümünde çalışan personelin listesi gibi. Sorgular SQL; ilerleyen bölümlerde detayları ile anlatılacaktır.

3.2. Veri Tipi (Data Type)

Bilgisayar, kayıtları tablolarda yapısal olarak tutarken, onların yapıları hakkında fikir sahibi olabilmek için bazı özelliklerinin önceden tanımlanması gerekir. Örneğin, personel sicil numarası alanının mutlaka bir tam sayıdan oluşacağı, personel ad ve soyadının harflerden oluşacağı, personelin çalıştığı bölümün harf ya da rakamlardan oluşacağı, personelin doğum tarihinin tarih bilgilerinden oluşacağı gibi. Bir veritabanı oluşturulurken, önce tablolar ve sonrada bu tablodaki her bir alanın veri tiplerinin ne olacağı tanımlanmak zorundadır. Bir tablo alanına veri girişi yapılmadan önce o alanın tamsayı mı yoksa harf mi; tarih mi yoksa ondalıklı bir sayı mı olacağı tanımlanmalı ve veriler daha sonra tabloya yazılmalıdır. Ayrıca, “bir alanın uzunluğu ne kadar olacak, harf girilebiliyorsa en fazla kaç harf girilebilecek?”, “rakam ise en fazla kaç basamaklı olabilir?” türünden soruları yanıtlamak için de yine VTYS bir alan için veri tipi belirlenmesini ister.

Her Veri Tabanı Programının veri tipleri farklıdır. Aşağıda Ms Access, MySQL ve Oracle veri tabanı programı örnek veri tipleri verilmiştir.

3.2.1. Access Veri Tabanı Veri Tipleri

Metin : Yazılacak bilgiler harflerden veya hem harf hem de sayılardan meydana geliyorsa kullanılacak veri türüdür. Bu alana boşlukta dahil olmak üzere en fazla 255 karakter bilgi yazılabilir. Bu alana yazılan bilgiler sadece sayılardan da oluşabilir, ama yazılan sayılar hesaplama işlemlerinde kullanılamazlar.

Not : Uzun metin yada metin ve sayı bileşimi kullanılabilir. Genelde açıklama ya da uzun bir not yazılacaksa bu alan kullanılır. 64.000 karakterle sınırlıdır.

Sayı : Öğrenci numarası, öğrencinin sınıfı gibi sayısal bilgiler için kullanılır. Sayısal alanlar matematiksel hesaplamalarda kullanılabilir. Borç, alacak, öğrenci harcı gibi.

Tarih/Saat : 100 ile 9999 arasındaki yıllar için tarih ve saat değerleridir.

Para Birimi : Bir ile dört arasındaki ondalık basamağı olan, matematik hesaplamalarında kullanılan para birimi değerleri ve sayısal veriler.

Otomatik Sayı : Tabloya yeni bir kayıt eklendiğinde, Access tarafından atanan benzersiz ardışık (biri birer artan) ya da rasgele sayılar. Otomatik sayı alanları değiştirilemez.

Evet/Hayır : Yalnızca iki değerden birini içeren alanlar Evet / Hayır, Doğru / Yanlış, Açık / Kapalı gibi alanlar gibi.

OLE Nesnesi : Access tablosuna bağlanmış ya da katıştırılmış bir nesne. (Microsoft Word veya Excel çalışma sayfası gibi)

Köprü : Tıklandığında kullanıcıyı başka bir dosyaya, dosyadaki bir konuma veya Internet'teki (www) bir bölgeye yönlendiren bağlantı.

Arama Sihirbazı : Değerleri başka tablo, sorgu ya da değerler listesindeki değerlerden seçilen bir alan yaratmamıza yardımcı olan sihirbaz.

3.2.2. MySQL Veri Tabanı Veri Tipleri

MySQL'de bir çok veri türü oluşturulabilir. Ancak Web programları açısından önemli olan bir kaç ve özellikleri şöyle sıralanabilir:

INT : Tamsayı: -2147483648'den 2147483647 kadar değişen diziye "signed" (işaretli), 0'dan 4294967295'e kadar değişenine "unsigned" (işaretsiz) denir.

VARCHAR(n) : n sayısını geçmemek şartıyla değişen boyutta karakter olabilir

CHAR(n) : Kesinlikle n sayısı kadar karakter olabilir.

TEXT : En fazla 65535(2¹⁶-1) karakter alabilen metin alanı.

MEDIUMTEXT : En fazla 16777215(2²⁴-1) karakter alabilen metin alanı.

DATE : 1000-01-01'den 9999-12-31'e kadar değişebilen tarih alanı.

TIMESTAMP : 1 Ocak 1970'den 18 Ocak 2038'e kadar olan ve Yıl+Ay+Gün+Saat+Dakika+Saniye biçimindeki zaman bilgisi.

3.2.3. Oracle Veri Tabanı Veri Tipleri

CHAR(sayı): Sabit uzunluktaki alfasayısal verilerin tutulabildiği alanlar için kullanılır. Oracle 7 ve daha önceki sürümler için bu alanın uzunluğu en fazla 255 karakter olabilir. Oracle 8 ve sonrasında 2000 karakter uzunluğundadır. Eğer, sayı ile ifade edilen numaradan daha kısa uzunlukta veriler girilirse Oracle kaydın sonuna boşluk ekleyerek sabit uzunluğa kadar getirir. Örnek char(20).

VARCHAR2(sayı): Değişken uzunluklu alfasayısal verilerin tutulduğu alanlar için kullanılır. Oracle 7 ve önceki sürümlerinde 2000 karakter, Oracle 8 ve sonraki sürümlerinde 4000 karakter uzunluğunda bilgi girilebilir. Örnek varchar2(30).

NUMBER(n,p): Tamsayı ve Gerçel sayılar için kullanılan sayısal veri tipidir. Tam kısım en fazla 38 basamak olabilir. Ondalık kısmın basamak sayısı da -84 ile 127 arasında değişmektedir. Number veri tipinden türetilmiş int[eger], dec[imal], smallint ve real veri tipleri de kullanılabilir.

DATE: Tarih tutan alanlar için kullanılır. Bu tip alanlarda, tarih bilgileri ve saat bilgileri tutulabilir. Tarih formatları Oracle yüklenilirken seçilen dile göre değişir. Amerikan standardı için 'DD-MON-YY' dir. Yani bir tarih '03-MAY-01' şeklinde görünür. NLS_DATE_FORMAT parametresi ile tarih formatı değiştirilebilir. Tarihsel alanlar üzerinde aritmetiksel işlemler yapılabilir. Sistem tarihi SYSDATE fonksiyonu kullanılarak öğrenilebilir. Sayısal veya karakter olarak tanımlı bir alandaki veriler TO_DATE fonksiyonu ile tarih tipine çevrilebilir.

LONG: 2 GB 'a kadar bilgi tutabilen karakter alanlar için kullanılır. Bir tabloda bu tipten ancak bir adet alan tanımlanabilir. Long veri tipine sahip alanlar için index oluşturulamaz.

Not: Oracle'da boolean veri tipi yoktur. Bunun için char(1) ya da number(1) şeklinde tanımlama yapıp kullanılabilir.

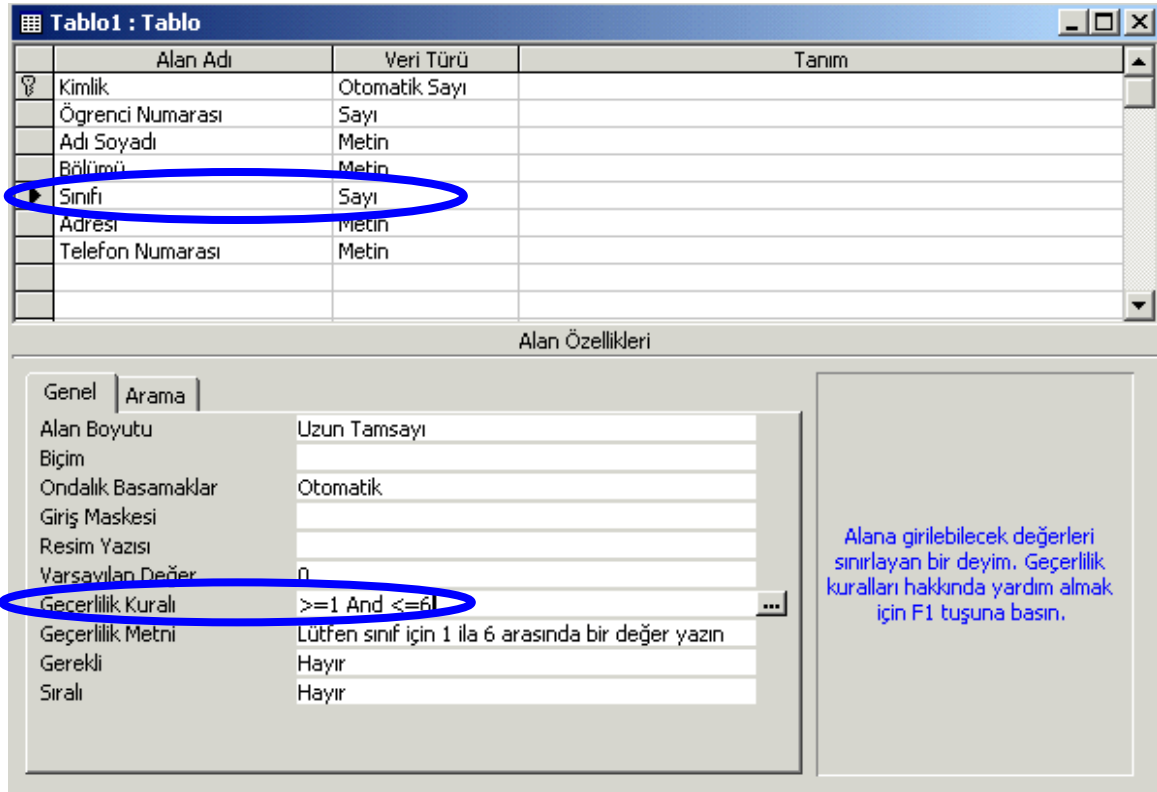
Not: Bir tablonun alanları kendi veri tipine uygun değerler alabildiği gibi bir de NULL değer alabilirler. NULL değeri sayısal olarak 0'dan ve karakter olarak ta boş karakterden(' ') farklıdır.

3.3. Zorlayıcı (Constraint)

Herhangi bir alan için girilebilecek verileri kısıtlayıcı kurallara zorlayıcılar denir. İlgili alana girilebilecek değerleri sınırlayan bir deyim yazılır. Kullanımı bazen çok faydalıdır ve özellikle yanlış bilgi girişini engeller ve verilerin doğru girilmesini zorunlu hale getirir. Kullanıcı, zorlayıcıda belirtilen kural dışında bir veriyi tabloya yazmaya çalıştığında, VTYs hata verir. Böylelikle veritabanına kullanıcının keyfi değerler girmesi önlenmiş olur ve veri tabanında tutarlılık sağlanmış olur.

Örneğin, bir öğrencinin sınıf bilgisine ait değerler yazılırken bu alan için rakamsal 1 ile 6 arasında bir zorlayıcı değer tanımlanırsa; veri girişi sırasında 1 ile 6 arasındaki değer dışında bir değerın sınıf bilgisi alanına yazılması engellenmiş olur. Dolayısı ile sınıf için yazılmaması gereken bir değer; bilgi girişi başlangıcında kontrol edilmiş olur.

Şekil 3.3.1. de mavi çizgilerle çevrelenmiş alanda Ms Access veri tabanı programında bir kısıtlayıcının tanımlanması yapılmış ve özellikle kullanıcı tarafından yanlış bilgi girişi engellenmiş ve verilerin doğru yazılması zorunlu hale getirilmiştir.



Alan Adı	Veri Türü	Tanım
Kimlik	Otomatik Sayı	
Öğrenci Numarası	Sayı	
Adı Soyadı	Metin	
Bölümü	Metin	
Sınıfı	Sayı	
Adresi	Metin	
Telefon Numarası	Metin	

Alan Özellikleri

Genel Arama

Alan Boyutu: Uzun Tamsayı

Biçim:

Ondalık Basamaklar: Otomatik

Giriş Maskesi:

Resim Yazısı:

Varsayılan Değer: 0

Geçerlilik Kuralı: >=1 And <=6

Geçerlilik Metni: Lütfen sınıf için 1 ile 6 arasında bir değer yazın

Gerekli: Hayır

Sıralı: Hayır

Alana girilebilecek değerleri sınırlandıran bir deyim. Geçerlilik kuralları hakkında yardım almak için F1 tuşuna basın.

Şekil 3.3.1. MS Access de tanımlanmış zorlayıcı ekranı

3.4. Anahtar (Key)

Anahtar bir veya birden fazla alanın bir satır için niteleyici olarak girilmesi için tanımlanan özel bir çeşit zorlayıcıdır. Tekrarlamayacak bir anahtar alan tanımlandığında, bu anahtar alana birincil anahtar alan denir. Primary Key, Unique Key ve Foreign Key olmak üzere 3 çeşit anahtar vardır.

Primary Key (Birincil Anahtar) : Bir tablodaki, her bir satırın yerine vekil olabilecek bir anahtar veridir. Tabloda bu alana ait bilginin tekrarlanmaması gerekir. Standart olarak bir tabloda verilerin, fiziksel hafıza üstünde de hangi alana göre dizileceğini de primary key belirler. Bu, bazen bir tek alan olabileceği gibi, bazen birden fazla alan da birleşerek bir birincil anahtar oluşturabilir. Örneğin programda personelin sicil numarası alanına göre aramalar yapılacaksa Primary key personel sicil numarası olmalıdır. Personelin ad ve soyadına göre aramalar yapılacaksa ad ve soyad alanları birleştirilerek iki alandan tek anahtar alan tanımlaması yapılır.

Unique Key(Tekil Anahtar): Unique Key olarak tanımlanan alan için bir değer sadece bir kere girilebilir. Bir başka satıra daha aynı verinin girilmesine izin verilmez. Primary Key 'den farklı olarak Unique Key, NULL (boşluk) değerini alabilir. Örneğin programda her personele ait bir sicil numarası olacağı için bu alan Unique key olarak tanımlanabilir. Ama isim alanı birden fazla aynı isme sahip personel olabileceği için bir Unique key olarak tanımlanamaz. Ali isimli birden fazla personel olabileceği gibi.

Foreign Key (Yabancı Anahtar) : Bir tabloya girilebilecek değerleri başka bir tablonun belli bir alanında yer alabilecek veri grubu ile sınırlandırmaya ve en önemlisi de ilişkilendirmeye yarar. Örneğin, olmayan bir kitabın ödünç tablosuna eklenememesi ve ödünç tablosuna eklenen bir kitabın numarası aracılığıyla detay bilgilerine erişilmesi gibi. Burada Kitap.KitapNo birincil anahtar alan; Odunc.kitapNo ise yabancı anahtardır.

3.5. Index (İndeks)

Kütüphanelerdeki kitapların raflardaki dizilişlerini ele alalım. Bir kitap arandığında, kitaplar bir kurala göre dizilmemişlerse, her bir kitaba teker teker bakılması gerekir. Kitaplar raflara alfabetik dizilirse, her bir kitap tek tek gözden geçirilmek zorunda kalınmaz. Aranılan kitap ile bakılmakta olunan kitabın isimleri karşılaştırılır, sağa ya da sola yönelip aramaya devam edilir. Aynı şekilde yazarlarına ya da kütüphane numarasına göre sıralanmış birer liste olursa, bu kriterlere göre de aranılan kitap kolayca bulunur. Veritabanlarında indeks oluşturularak, veriler veritabanındaki kayıtlı oldukları sıradan başka bir sırada gösterilebilir ve tıpkı kütüphanedeki bir kitaba ulaşmada olduğu gibi istenilen veriye daha kısa sürede ve kolayca ulaşılabilir.

Temelde indekslerin ilişkisel veritabanında şu üç işlevi vardır:

1. Tekil indeksler, veri ilişkilerini ve veri bütünlüğünü sağlayan birincil anahtar alanlar oluşturmada kullanılır.
2. İndeks olan alanın değerine göre bir kaydın kayıtlar arasındaki sırasını gösterirler.
3. Sorguların neticelenme sürelerini kısaltırlar.

Constraint'ler (zorlayıcılar) aslında index'lere benzerler ama indekslerden farklı olarak bir tek tablo üstünde etkili olmayabilirler. Özellikle yabancı anahtar zorlayıcısı ilişkisel veri girişi için oldukça etkili bir zorlayıcıdır. Ancak bir Foreign Key tanımlı yapabilmek için, FOREIGN KEY yabancı anahtarının asıl tablosunda birincil anahtar olması gerekir.

3.6. View (Görüntü)

Bazen, tabloları olduklarından farklı gösterecek filtreleme ihtiyacı duyulur. Bu türden işlevler için VIEW kullanılır. VIEW 'ler, saklanmış sorgulardan ibarettirler. Aslında tablo gibi kullanılsa da halihazırda böyle bir tablo veritabanında bulunmaz, sadece view(görüntüsü) bulunur. VIEW 'ler şu görevler için kullanılır:

- * Kullanıcıların bazı kritik tabloların sadece belli sütunlarını veya satırlarını görmesi istenildiğinde,
- * Kullanıcıların, çeşitli birim dönüşümlerinden geçmiş değerler görmeleri gerektiğinde,
- * Halihazırdaki tablolarda var olan verilerin başka bir tablo formatında sunulması gerektiğinde
 - Çok kompleks sorguları basitleştirmek için

Örneğin Kitap tablosunda sadece Bilgi Teknolojileri türündeki kitapların yer alacağı bir VIEW şu şekilde oluşturulabilir:

```
CREATE VIEW view_adi [(kolon1,  
kolon2...)] AS  
SELECT tablo1.kolon_adi_1, tablo2.kolon_adi_1  
FROM tablo_adi_1, tablo_adi_2 ;
```

Not : MSAccess 'de VIEW oluşturulamaz. SQLServer, SyBase, Oracle gibi orta ve büyük ölçekli VTYS 'lerde oluşturulabilir.

3.7. Joining (ilişkilendirme)

İki veya daha fazla tabloyu birlikte sorgulama işlemine join ismi verilir. İlişkisel veritabanının en temelinde birden fazla tablo üstünde birlikte işlem yapabilmek yatar. Bu sayede verilerin tekrarlanması önlenmiş olur ve sonuçta veri yönetimi kolaylaşır. Örneğin, Kitap tablosunda, Kitabın bir tekil numara ile listesini tutmak ve ödünç listesinde de bu Kitabın kim tarafından alındığının, geri getirilip getirilmediğinin kaydı tutulmaktadır. Bazen, bu iki tablodaki bilgilere de bir tek sorgu sonucu olarak ihtiyaç duyulabilir.

Örneğin; elimizde öyle bir sonuç olmalıdır ki, hangi kitabın kim tarafından ödünç alındığını bir listede görme ihtiyacı duyulsun. Bu iki tablo birbirine, kitapNo alanı ile bağlıdır. Çünkü, ödünç verilen bir kitap hakkında detaylı bilgi edinilmek istenildiğinde, ödünç listesinden kitap numarasını alıp, daha sonra Kitap tablosundan aynı numarayı bulmak ve karşılığındaki kitap hakkındaki detayları görmek.