**VERİ MODELLERİ**

Veri tabanında depolanan verilerin mantıksal bir düzeyde düzenlemek için kullanılan kavramlar, yapılar ve işlemlerin tamamına veri modeli denilmektedir. Her VTYS belirli bir veri modeli kullanır. Seçilen veri modeline göre veriler arasındaki ilişkiler belirlenir. Çoğunlukla kullanılan 4 veri modeli aşağıdaki maddeler halinde açıklanmıştır.

* **Hiyerarşik Veri Modeli**

İlk kullanılan veri modeli olan hiyerarşik veri modeli 1960 ve 1970 li yıllarda kullanılmıştır. Veriler bir ağaç şeklinde tutulur .Bu ağaçta veriler tek bir üst dala sahip fakat birden fazla alt dalı olabilir. Bu modele Sıradüzensel Veri modeli de denilmektedir.

\*\*Varlıklar arası ilişkiler kurulurken her varlı sadece bir varlıkla ilişki kurabilmektedir. Bir varlığın birden fazla ilişkisi söz konusu ise tekrar oluşturulması gerekmektedir buda gereksiz veriye sebep olacaktır.

* **Ağ Veri Modeli**

Sıradüzensen veri modelinin eksikliklerinden yola çıkarak geliştirilmiş tablo ve grafik temelli bir veri modelidir. Bu modelde bir veri birden fazla veri ile ilişki kurabilmektedir ve ilişki sayısında bir sınırlama bulunmamaktadır. Bu modele Şebeke veri modeli de denilmektedir. Bu modelde ağlar arası ilişki kurulamaması bir eksiklik olarak görülmektedir.

****

* **İlişkisel Veri Modeli**

1985 yılından itibaren yaygın kullanıma başlanmış bir veri modeli türüdür. Günümüzde kullanılan çoğu veri tabanı yönetim sistemi ilişkisel veri tabanı modelini kullanmaktadır. Bu modelde veriler tablolar halinde tutulurlar ve veriler kendilerine ayrılmış alanlarda tutulurlar. Aynı alanda bulunan veriler aynı tür veriyi ifade ederler. Örneğin Öğrenci numarası alanı altında bulunan bütün satırlar öğrenci numarası bilgisini tutmaktadırlar.

İlişkisel veri modelinde verilerin tutulduğu tablolar bir birleri ili ilişki kurabilmektedir. Bu ilişkileri birincil anahtar ve yabancı anahtarlar aracılığı ile yapılmaktadır.

****

İlişkisel veri modelinde tablolar arasındaki ilişkiler tabloların içerdiği verilere ve tabloların yapışana göre değişiklikler gösterebilir. Temel olarak 4 çeşit ilişki kurma yöntemi bulunmaktadır. Bunlar şöyle özetlenebilir;

* + - **Bire-Bir ilişkiler:**

İlişi kurulan tablolar da bulunan alanlardaki verilerden aynı değerleri içerecek veriler bulunmadığı durumlarda bire-bir ilişki kurulabilir. İlişkili tablolardan birinin birincil anahtar alanı ile diğer tablonun ilişkili veri içeren herhangi bir aranı ilişkilendirilir.

* + - **Bire-Çok İlişkiler:**

İlişki kurulacak tablolardan birinin birincil anahtar değeri diğer tablodaki alanda birden fazla alanla ilişkilendirilmesi durumunda oluşan Bire –çok ilişkilendirme yapılır.

* + - **Çoğa-Çok ilişkiler:**

İlişki kurulan tabloların her ikisinde de birden fazla değer diğer tablonun birden fazla alanına denk geliyorsa bu durumda Çoğa-Çok ilişkilendirme kurulmuş olur.

* **Nesneye Yönelik Veri Modeli**

Nesne tabanlı programlamada kullanılan kalıtım ve miras yapısı bu modelde uygulanmaktadır.1990 yıllarından itibaren kullanılmaya başlanılan bir modeldir. İlişkisel modelde kullanılan ilişki kavramı bu modelde sınıf kavramına denk gelmektedir. ilişkisel modele göre daha esnek ve kullanışlı bir yapıya sahiptir.

**İLİŞKİSEL VERİ TABANI TASARIMI**

* **Veri Tabanı Tasarımı**

Bir proje başlamadan önce veri tabanı tasarımının doğru bir şekilde yapılması gerekmektedir. Veri tabanı tasarımının tutarlı bir şekilde yapılmaması durumunda projenin ilerleyen safhalarında karmaşıklıklar çıkacak ve bu karmaşıklıkların çözümü çok daha zor olacaktır. Bir veri tabanı tasarımı yapılırken izlenmesi gereken adımlar şöyle özetlenebilir.

i-Veri tabanında tutulacak veriler belirlenerek gruplandırılmalıdır.

ii-Belirlenen veriler ve alan adları doğrultusunda tablolar oluşturulmalıdır. Tablolar oluşturulurken isimlendirmelerde kullanılan harf ve karakter kısıtlamalarına dikkat edilmelidir.

iii- Tablolarda bulunan kayıtların bir birlerinden ayırt edilebilmesini sağlamak için anahtar alan belirlenmelidir.

iv-Oluşturulan bir tabloda veriler tekrar ediyor ise bu tekrardan kaçınmak için ayrı bir ara tablo oluşturularak bu tekrardan kaçınılmalıdır.

* **İlişkisel Veri tabanı Kavramları**

Veri tabanında tutulacak bilgilerin daha üst seviyede gösterilmesi için varlık-ilişki modeli kullanılır. Veri tabanı yönetim sistemlerinde kullanılmayan bu model çoğunlukla kavramsal olarak veri tabanı tanımlamasında kullanılmaktadır.

* + **Varlık**

Modelleme yapılırken kullanılan en temel öğedir. Var olan ve benzerlerinden ayırt edilebilen bir bir öğeyi varlık olarak ifade edebiliriz. Örnek olarak öğrenci, müdür, okul gibi öğeler birer varlık olarak ifade edilirler. Birden fazla varlığın bir araya gelerek oluşturdukları kümeye varlık kümesi adı verilir.

Varlık

* + **Nitelik**

Her bir varlık kendine ait özelliklere sahiptir. E-R diyagramlarında bu özellikler nitelik olarak isimlendirilmektedirler. Nitelikler şematik olarak gösterildiğinde oval bir şekil kullanılarak gösterilmektedirler. Özellikler arasında anahtar değer taşıyan bir özellik bulunuyor ise buna anahtar nitelik denir ve altı çizili bir şekilde gösterilir.

****

Yukarıdaki şekilden görüldüğü üzere varlık dikdörtgen şeklindeki ve nitelikler ise ova bir şekilde temsil edilmektedir. Yukarıda gösterilen nitelik değerleri sadece tek bir değer alabildiğinden dolayı tek değerli nitelik olarak isimlendirilmektedirler. Varlık niteliklerinden herhangi biri birden fazla değer alabiliyorsa bu nitelik çok değerli nitelik olarak isimlendirilir ve aşağıdaki şekilde gösterilir.

****

Öğrenci Adında bir varlık olduğunu düşünelim ve {ogr\_no, ogr\_adi, ogr\_bolum) niteliklerine sahip olduğu nu varsayalım. Bu durumda Öğrenci varlığının E-R diyagramı Aşağıdaki şekilde olacaktır.

****

* + **Etki Alanı**

E-R Şemasında herhangi bir gösterimi olmayan etki alanı özelliği ,bir varlığa ait olan niteliklerin alabileceği değer sınırını belirtmektedir.

* + **İlişki**

Farklı varlıklar arasında ilişki kurma ihtiyacı olduğu zaman kullanılan bir lidir. Baklava dilimi şekline sahip ve varlıklar arasın konularak içerisine ilişki ismi yazılır.

****

Varlıklar arasında bire bir ve bire çok ve çoğa çok ilişkiler kurulabilir.

**1-n ilişki örneği;**

****

Bir Hocanın birden fazla ders verebileceğine dair ilişki modeli

* 1. **ilişki örneği;**

****

Bir Hocanın sadece bir okulda Çalışabileceğini gösteren ilişki modeli.

**n-n ilişki örneği;**



Çok sayıda öğrenci çok sayıda sınava girebileceğini gösteren ilişki modeli.

* + **Zayıf Varlık**

Oluşturulan bir varlık kümesi anahtar değere sahip ise güçlü varlık kümesi olarak tanımlanmaktadır. Varlık kümesi anahtar bir niteliğe sahip olmadığı durumlarda zayıf varlık olarak isimlendirilmektedir. Zayıf varlık kümeleri güçlü varlık kümeleri ile ilişkilendirilerek varlıklarını devam ettirirler. Zayıf varlık kümeleri çift çizgili dikdörtgen şeklinde gösterilmektedirler.

**1 n**

Sahiptir

Hoca

Bölüm

Varlık ilişki modellerinde kullanılmakta olan semboller aşağıdaki tabloda gösterilmektedir.

|  |  |
| --- | --- |
| **Gösterim Şekli** | **Anlamı** |
|  | **Varlık** |
|  | **Zayıf Varlık** |
|  | **Nitelik** |
|  | **Çok Değerli Nitelik** |
|  | **Anahtar Nitelik** |
|  | **İlişki** |

* **Varlık ilişki modelleri ve tablo tasarımında kullanımı**

Varlık ilişki modellerinin veri tabanı yönetim sistemlerinde kullanılabilmesi için tablo tasarımı haline dönüştürülmesi gerekmektedir. İlişki modelinden tablo tasarımına geçiş esnasında varlık değerleri birer tabloyu nitelikler ise oluşturulan tabloların sütün alanlarını ifade eder.

ER modelinden tablo tasarımına geçişte temel olarak uygulanması gereken adımlar aşağıdaki şekilde gösterilmektedir.

1. **Her varlık kümesine karşılık bir tablo oluşturulur**

|  |
| --- |
| **Öğrenci Tablosu**  Öğrenci |
|  |

1. **Varlıklarda bulunan nitelikler tablolarda Sütünlara karşılık gelmelidir.**

|  |
| --- |
| **Öğrenci Tablosu**  Öğrenci |
| **Adi** |
|  |

1. İlişki belirtmede kullanılan alan birinci varlık kümesinin birincil anahtarı diğer varlık kümesinin ise yabancı anahtarı olarak belirtilmelidir**.**

|  |
| --- |
| **Öğrenci Tablosu**  Öğrenci |
| **Ogr\_no** |
| **Adi** |

1. Oluşturulan ER modelinde varlıklar arasında kurulan ilişkilerde tanımlayıcı nitelikler var ise tablolara yabancıl anahtar sütunu olarak eklenir**.**

|  |
| --- |
| **Öğrenci Tablosu**  Öğrenci |
| **Ogr\_no** |
| **Adi** |
| **Tarih** |

İlişki

Örnek; Yukarıda belirtilen adımlar doğrultusunda aşağıda belirtilen varlık ilişki modelinin tablo dönüşümünü görelim.

****

1.Adım: Personel ve Bölüm adında birer tablo oluşturulur.

2.Adım: Personel tablosunda {id, adi, adres,maas) sütunları ,bolum tablosunda ise {B\_id , B\_Adi) sütunları bulunacaktır.

3.Personel tablosunun birincil anahtarı olan id değeri bölüm tablosuna yabancı anahtar olarak eklenmelidir.

|  |
| --- |
| **Bölüm Tablosu** |
| **B\_id**  **B\_adi**  **id** |

|  |
| --- |
| **Personel Tablosu** |
| **İd**  **Adi**  **Adres**  **maaş** |