**Veritabanı İlişki Tipleri**

Depolanması gereken verilerin gittikçe artması ve hızın önem kazanması ile Oracle, MsSQL ve MySQL gibi ilişkisel veritabanlarında (relational database) tasarım oldukça önemli bir yer tutmaktadır. Bu nedenle veritabanı kurulurken ilişki tiplerine dikkat edilmelidir.

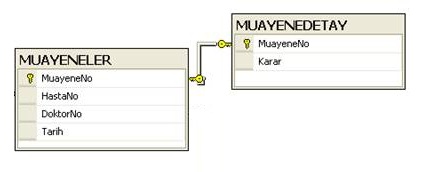
İlişkisel veri tabanlarında 3 tip ilişki vardır:

* Bire Bir
* Bire Çok
* Çoka Çok

**Birincil Anahtar (Primary Key):** Tablolarda benzersiz kayıtlar elde etmemizi sağlayan sütuna verilen addır.

**İkincil Anahtar (Foreign Key):** Bir tabloda benzersiz kayıt oluşturmayı sağlayan sütunun diğer tabloda bir sütun olarak bulunmasına denir.

**Bire Bir İlişki Tipi:** İki tablo arasındaki ilişkiyi sağlayacak olan ortak alanların ikiside birincil anahtar (primary key) ise birebir ilişki olur.



Yukarıdaki şekilde MUAYENELER ve MUAYENEDETAY adlı iki tablo vardır. Her muayeninin bir detayı olacağı ve her detayında sadece bir muayene için olacağını düşünürsek güzel bir bire bir ilişki tipi olur. Burada ilişki iki tablonun MuayeneNo adlı birincil anahtarı ile sağlanır.

Eğer MUAYENELER ve MUAYENEDETAY tablolarını birleştirerek veritabanını sorgulamak istersek aşağıdaki SQL sorgusunu çalıştırmamız yeterli olacaktır:

|  |  |
| --- | --- |
|  | **SELECT** \* **FROM** Muayene, MuayeneDetay  **WHERE** Muayene.MuayeneNo = MuayeneDetay.MuayeneNo |

**Bire Çok İlişki Tipi:** İki tablo arasındaki ilişkiyi sağlayacak olan ortak alanlardan birisi birincil anahtar (primary key) diğeri yabancı anahtar (foreign key) ise bire çok ilişki olur.

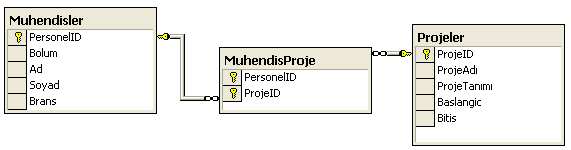


Yukarıdaki şekilde MUAYENELER ve HASTALAR adlı iki tablo vardır. Her hastanın birden fazla kez muayne olabileceğini ve her muayenenin yalnız bir hastaya ait olduğunu düşünürsek güzel bir bire çok ilişki tipi olur. Burada ilişki HASTALAR tablosundaki birincil anahtar HastaNo ile MUAYENE tablosundaki yabancı anahtar HastaNo ile sağlanır.

Eğer MUAYENELER ve HASTALAR tablolarını birleştirerek veritabanını sorgulamak istersek aşağıdaki SQL sorgusunu çalıştırmamız yeterli olacaktır:

|  |  |
| --- | --- |
|  | **SELECT** \* **FROM** Muayene, Hastalar  **WHERE** Muayene.HastaNo = Hastalar.HastaNo |

**Çoka Çok İlişki Tipi:** Çoktan çoğa ilişikiler karmaşıklığı nedeniyle veritabanında bir çok soruna neden oldur. Sorunun temel nedeni bir tablodaki anahtar ilişkili olduğu tabloda birden çok kez olabilirken karşıtı da doğrudur.  
Bu yüzden istenilen ilişkiselliği sağlamak için ilişkiyi birden çoğa indirgemek gerekir. Yani araya fazladan bir tablo koyarak karmaşıklıktan kurtulmuş oluruz.



Yukarıdaki tablolardanda görüldüğü gibi eğer ortadaki tablo olmasaydı birçok mühendisle bir çok proje ilişkisi oluşacak ve karmaşıklık yaratacaktı. Fakat araya yeni bir tablo ekleyerek MUHENDISLER ve MUHENDISPROJE tabloları arasında bire çok ayrıca MUHENDISPROJE ile PROJELER tablosu arasında bire çok bir ilişki oluşturuldu.

Böylece verimli bir ilişkisel veritabanı oluşturuldu.

Çoktan çoğa ilişkiyi birden çoğa indirgediğimiz için iki tane birden çoğa ilişki elde ettik yine bu sebepten iki tane ikincil anahtar (Foreign Key) ortaya çıktı. Dolayısıyla SQL sorugusu iki adet olacak. Birincisi bir mühendisin yaptığı proje, ikicisi ise bir projede görevli mühendis olacak.

Birinci sorgumuz (Id numarası 1 olan mühendisin yaptığı projeleri getiren sorgu)

|  |  |
| --- | --- |
|  | **SELECT** \* **FROM** Muhendisler, MuhendşsProje, Projeler  **WHERE** MuhendisProje.PersonelID = Muhendisler.PersonelID **AND** Muhendisler.PersonelID=1 |

İkinci sorgumuz (Id numarası 1 olan projede görev alan mühendisleri getiren sorgu)

|  |  |
| --- | --- |
|  | **SELECT** \* **FROM** Muhendisler, MuhendşsProje, Projeler  **WHERE** MuhendisProje.ProjeID = Projeler.ProjeID **AND** Projeler.ProjeID=1 |