

# Veri Bilimciler için Veritabanı Eğitimi

# Kursa Başlamadan Önce

- Bu kurs tamamlandığı takdirde giriş düzeyi veri tabanı işlemlerini için güçlü bir temel oluşturabilecek genel bilgileri edinmiş olacaksınız
- Spesifik konular anlaşılması zor ve kişide ders esnasında mantığın oturması kolay olmayacağından bol bol bireysel pratik gerekmektedir
- Slaytlar yazılara boğulmadan görsellerle anlatılacaktır. Bu yüzden ders esnasında not tutulması **son derece** önemlidir
- Konu başlıkları temel düzey algoritmalar için yeterli olduğundan başlıklar araştırılmalı, bol bol uygulama ve teorik bilgiler içeren sitelerde araştırma yapılmalıdır
- Bir Veri bilimcisi, Yapay Zeka Mühendisi vb. ünvanlara sahip kişiler için güçlü bir yetenek oluşturmak için ilk adımı atabilirsiniz

# Ömer Cengiz

Github: [omercengiz](#)

Linkedin: [omercengiz96](#)

Twitter: [omerrcengizz](#)

# Veri Tabanı Nedir ve Neden İhtiyaç Duyarız?

## Veri Tabanı Nedir?

- İçerisinde verilerin belirli kurallara göre saklandığı yapılardır
- İçinde bulunan verilere “Veritabanı Yönetim Sistemleri” (Database Management Systems) ile ulaşılabilir/veri & veri tabanı üzerinde birtakım işlemler gerçekleştirilebilir

## Neden Veri Tabanına İhtiyaç Duyarız?

- Veri setimizin çok büyük olması durumunda
- Yanlış veri girilmesini önlemek istediğimizde
- Hassas verilerin saklanması
- Aynı verinin birden fazla kopyasının olmasını istemediğimizde

# Veri Tabanı Yönetim Sistemleri (DBMS) Nedir?

Veri Tabanı Yönetim Sistemleri yani kısaca DBMS'ler:

- Bir veya birçok veritabanını yönetmek veya analiz etmek için kullanılan yazılımlardır
- Lokal bilgisayarlar veya uzak sunucular üzerine kurulur
- Veri tablolarının oluşturulması (create), sorgulanması (query), verilerin güncellenmesi (update) ve silinmesi (delete) gibi birçok yönetim eylemlerinin gerçekleştirilmesini sağlar



# Çeşitli DBMS Tipleri

- İlişkisel Veri Tabanı Yönetim Sistemleri (Relational DBMS)
- Hiyerarşik Veri Tabanı Yönetim Sistemleri (Hierarchical DBMS)
- Belge Veri Tabanı Yönetim Sistemleri (Document DBMS)
- Ağ Modeli Veri Tabanı Yönetim Sistemleri (Network DBMS)
- Nesneye Yönelik Veri Tabanı Yönetim Sistemleri (Object-Oriented DBMS)
- NoSQL Veri Tabanı Yönetim Sistemleri (NoSQL DBMS)

# İlişkisel Veri Tabanları

Relational Databases

# İlişkisel Veri Tabanları Nedir?

- Verilerin, birbiriyle belirli bir ilişki içerisinde olan tablolarda saklandığı yapılardır
- Tablolar, veri tabanını oluşturan en basit yapılardır. Her bir tablo, satır ve sütunlardan (kayıtlar ve alanlar) oluşur
- Her bir sütunda yapılan kayda ait ilgili özelliğin verisi yer alır
- Her bir sütuna özel bir ad verilir & uygun veri tipi belirtilir

ID	First_Name	Last_Name	Age	City
1	Bill	Smith	22	London
2	Mary	Davis	53	New York
3	Martin	Green	27	Sydney
4	Rob	Lee	35	Geneva
5	Maria	Wilson	42	Paris

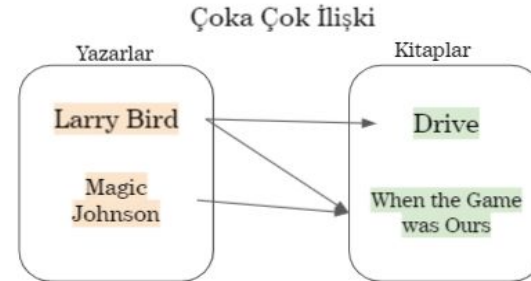
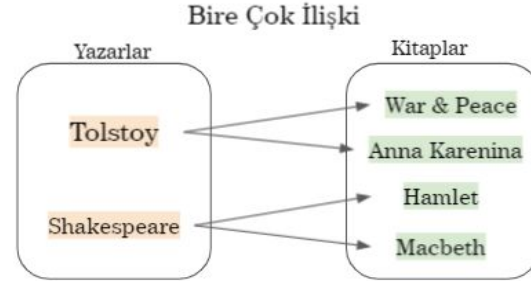
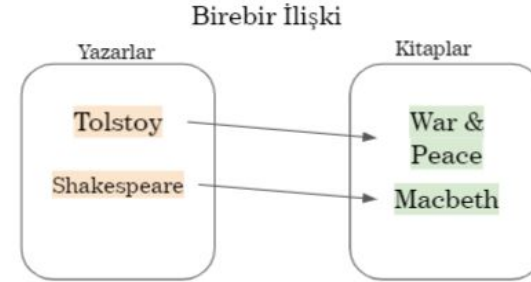
## 3 farklı sütun tipi bulunmaktadır:

- **Single:** içerisinde tek bir değer bulundurmaktadır.
- **Composite:** içerisinde birkaç değerden oluşan bir değer barındırır. (örn: isim soyisim sütunu verilerinin isim-soyisim kombinasyonlarından oluşması)
- **Multi-valued:** Bir kayıt için farklı değerler bulundurmaktadır. (bir ürünün renk bilgisinin “kırmızı, siyah” şeklinde girilmesi gibi)



# İlişki Tipleri

- **Bire Bir (One-to-One) (1:1):** Bir tablodaki bir kayıt, başka bir tablodaki bir ve yalnızca bir kayıtla ilişkilendirilir
- **Bire Çok (One-to-Many) (1:M):** Bir tablodaki bir kayıt, başka bir tablodaki birden fazla kayıtla ilişkilendirilir
- **Çoka Çok (Many-to-Many) (M:N):** Bir tablodaki birkaç farklı kayıt, başka bir tablodaki birkaç farklı kayıtla ilişkilendirilir.



# Anahtar Tipleri: Birincil Anahtar (Primary Key)

- İçerisindeki her bir değeri eşsiz olan ve bir tablodaki her bir satırı tanımlamızı sağlayan sütundur
- Ya bir tabloda önceden mevcut olur ya da kullanılan veri tabanı yönetim sistemi tarafından “auto-increment” edecek şekilde oluşturulur
- Her bir tablonun tek bir adet “birincil anahtar”ı bulunmak zorundadır
- Kompozit Birincil Anahtar: Birden fazla sütundan oluşan birincil anahtardır. “Ara Tablo (intermediate/linking/joining table)” nun oluştuğu durumlarda görülür

**Not:** Her ne kadar tabloda bulunan sütunlardan biri birincil anahtar olarak seçilebiliyor olsa da eğer kimlik numarası, kredi kartı numarası veya şifre gibi hassas verilerin sütunları birincil anahtar olarak seçilmemelidir.



# Eşsiz ve Yabancı Anahtar (Unique & Foreign Key)

## Eşsiz Anahtar:

- Tabloda, içerisindeki her bir değeri birbirinden farklı (eşsiz) olan sütundur
- Bir tabloda birden fazla eşsiz anahtar olabilir

roll_number	name	batch	phone_number	citizen_id
-------------	------	-------	--------------	------------

↑  
unique key

## Yabancı Anahtar:

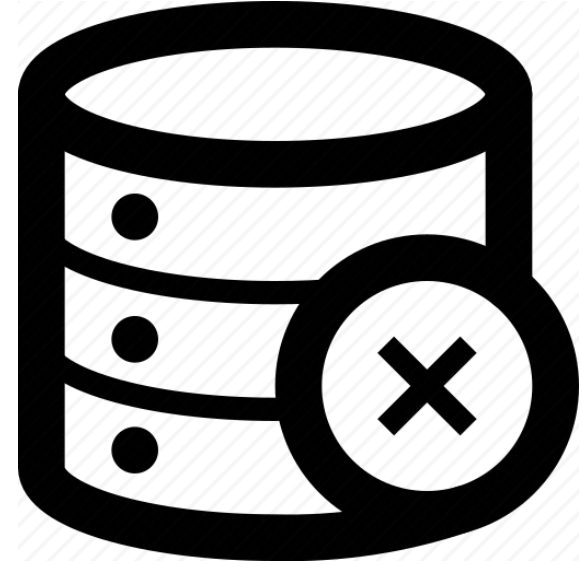
- İki tablo arasında ilişki kurmak için kullanılan sütun(lar)dır
- Referans anahtar olarak da bilinir
- Yabancı anahtarın değerleri, ilişki kurulacak olan tablonun “birincil anahtarı”nın değerleriyle aynıdır
- İlişki kuracak olan tablolardan birinin “primary key”i diğerinde “foreign key” sütunu olarak bulunur

# Neden İlişkisel Veri Tabanı Daha Yaygındır?

- Günümüzde en yaygın kullanılan veri tabanı tipidir
- İmplementasyonu kolay ve güçlü bir yapıya sahiptir
- Farklı büyüklükteki kurumlar tarafından kullanılabilir
- Verilerin güvenli, kurallı ve tutarlı bir şekilde yönetilmesini sağlar

Sık kullanılan ilişkisel veri tabanı yönetim sistemleri:

- Oracle, PostgreSQL, MySQL, SQLite, vs.



# Veri Tabanında Görülen Bazı Anomali (Hata) Tipleri

**Ekleme Anomalisi (Insertion Anomaly):** Veri tabanına yeni bir kayıt eklenirken görülen hatalardır

*Örn: Yeni bir kayıt açılırken “primary key” kısmının boş bırakılması (NULL gözükməsi)*

**Değiştirme Anomalisi (Modification Anomaly):** Veri tabanındaki bir verinin güncellenmesi sırasında görülen hatalardır

*Örn: Birden fazla yerde tekrarlanmış bir verideki değişikliğin her bir veri noktasına yansıtılmaması*

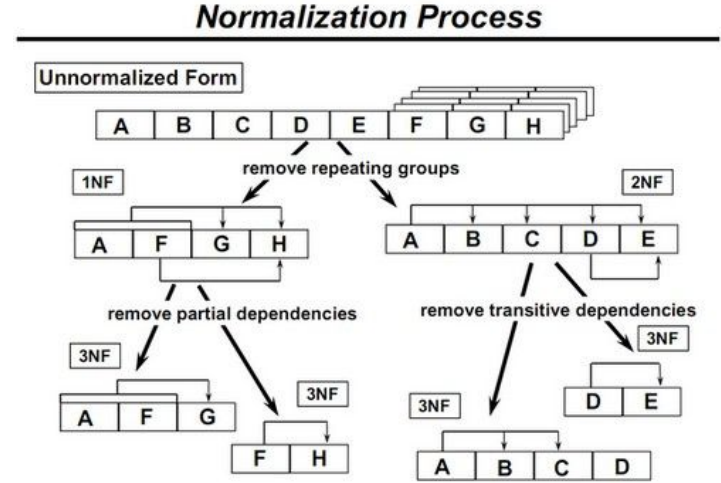
**Silme Anomalisi (Deletion Anomaly):** Bir verinin silinmesi sırasında ortaya çıkan hatalardır

*Örn: Bir kaydın silinmesiyle sütunlarında yer alan, işe yarayabilecek verilerin de kaybedilmesi*



# Normalizasyon Nedir?

- **Normalizasyon**, veri tabanlarında görülen hataları minimum'a indirmeye ve gereksiz veri tekrarının önüne geçmeye çalışan işlemdir
- Normalizasyon uygulanmış veri tabanlarında **arama algoritmaları** (searching algorithms) daha hızlı çalışır, böylelikle istenilen veriyi bulmak kolaylaşır
- Veri tabanının daha az yer kaplamasını sağlar, saklama alanı daha verimli kullanılır
- Normalizasyon, “Normalizasyon Formları” olarak da bilinen seviyeli kurallar bütünüyle uygulanır



# ACID

Gerçekleştirilecek olan her bir “transaction”un **ACID (Atomic, Consistent, Isolated, Durable)** olarak da bilinen 4 niteliği sağlaması gerekir.

**Atomiklik (Atomicity):** Bir transaction başarılı bir şekilde tamamlanabiliyorsa gerçekleştirilir, eğer herhangi bir noktada hata çıkıyorsa hiç gerçekleştirilmez ve veri tabanı transaction’dan önceki haline geri döner.

**Tutarlılık (Consistency):** Veri üzerinde yapılacak bir işlemin veri tabanındaki herhangi bir kuralı ihlal etmemesi gerektiğini ifade eder. *Örn: bir kolona belirlenen veri tipi dışında başka bir veri tipi girilememesi.*

**Izolasyon (Isolation):** Veri tabanı yönetim sistemleri transaction’ları belirli bir sırayla gerçekleştirir, bu durum aynı kayıt üzerinde gerçekleştirilen işlemlerin birbiriyle çakışmasını önler. *Örn: Stoktaki bir ürünün aynı anda iki farklı kişi tarafından sepete atılamaması.*

**Durability (Dayanıklılık):** Elektrik kesilmesi veya diskte çıkan herhangi bir sorun çıkarsa dahi tamamlanmış bir transaction’un sonradan kaybolmayacağını ifade eder.

Çoğu veri tabanı yönetim sistemleri, gerçekleştirilen transaction’ları “**transaction log**” denilen işlemler geçmişine not eder. Bu sayede sistem güncel durumunu korur.

# Kümelenmiş ve Kümelenmemiş Index

- Verilerin ana tabloda fiziksel olarak depolanma sırasını tanımlar
- Tablo verileri yalnızca bir şekilde sıralanabilir, bu nedenle tablo başına yalnızca bir tane clustered index olabilir
- Çoğu DBMS'lerde Primary Key için otomatik olarak clustered index oluşturulur
- Primary Key dışında bir sütun üzerinden oluşturulan index tablosudur
- Non-clustered index tablolarında belirtilen sütun ve ana tablodaki primary key sütunu bulunur
- Index tablosu Binary search yöntemiyle okunur, sonra sorguda bahsedilen koşulları sağlayan kayıtların diğer gerekli bilgilerine ana tablodan ulaşılır
- Bir tablonun tek bir tane clustered index'i bulunabilirken birden fazla unclustered index'i olabilir



# SQL

Structured Query Language

# SQL Nedir ve Temel SQL Syntax'ı Nasıldır?

## SQL Nedir?

- Yönetim sistemi tarafından veri tabanı oluşturma, verileri okuma, veriyi güncelleme ve silme gibi işlemleri yapmamızı sağlayan sorgulama dilidir
- Sanılanın aksine, SQL bir yazılım dili değildir
- RDBMS'ler için standard dil olarak kabul edilir

## Temel SQL Syntax'ı Nasıldır?

- Bir SQL ifadesi SELECT, INSERT, UPDATE, DELETE gibi bir anahtar kelimeyle başlar ve noktalı virgül ";" ile sonlanır
- Anahtar kelimeler genelde büyük harflerle yazılır
- Yorumlar, başına "--" konularak yapılır

```
SELECT column1, column2, ... column FROM table_name;
```