Database bir veriyi kalıcı olarak saklamamızı sağlar.

Genel olarak databaseler sql ve nosql tabanlı olarak 2 başlık altında ele alınır

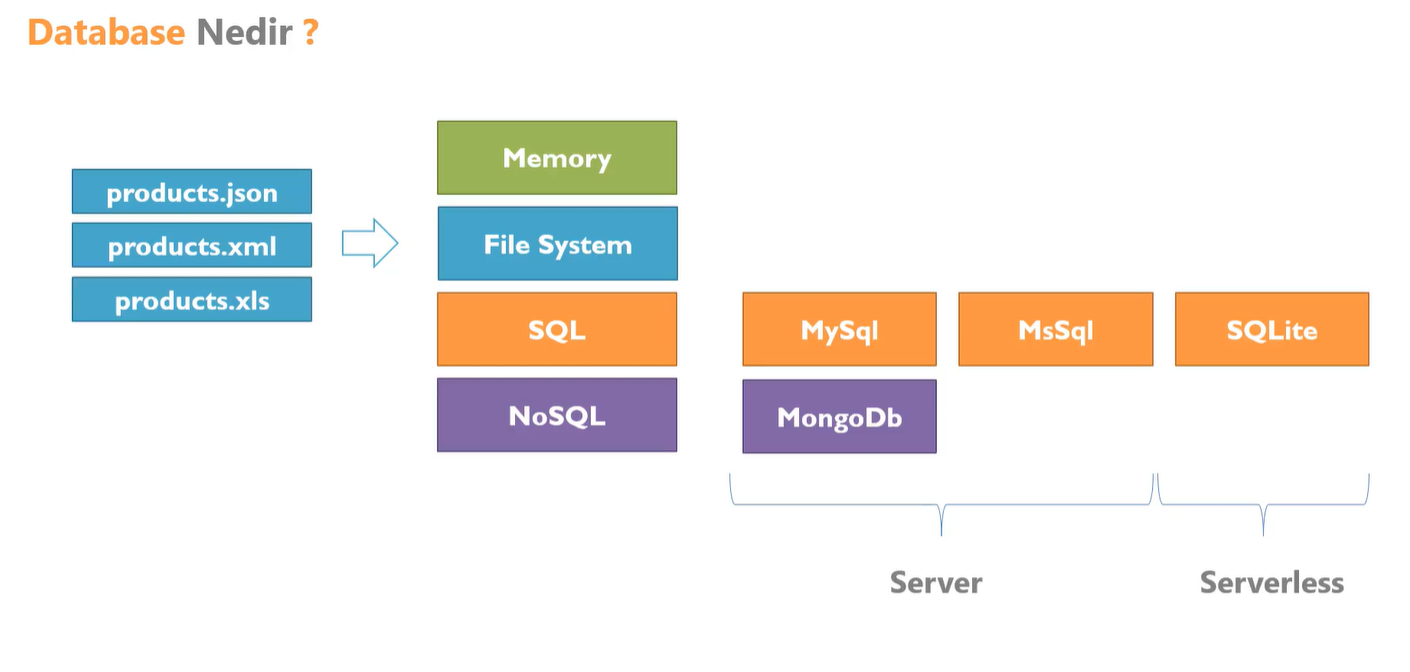
Sql veri tabanları = Mysql, Mssql, Sqlite

Nosql = mongodb

Sql ve Nosql veri saklama yapısına göre birbirinden farklıdır

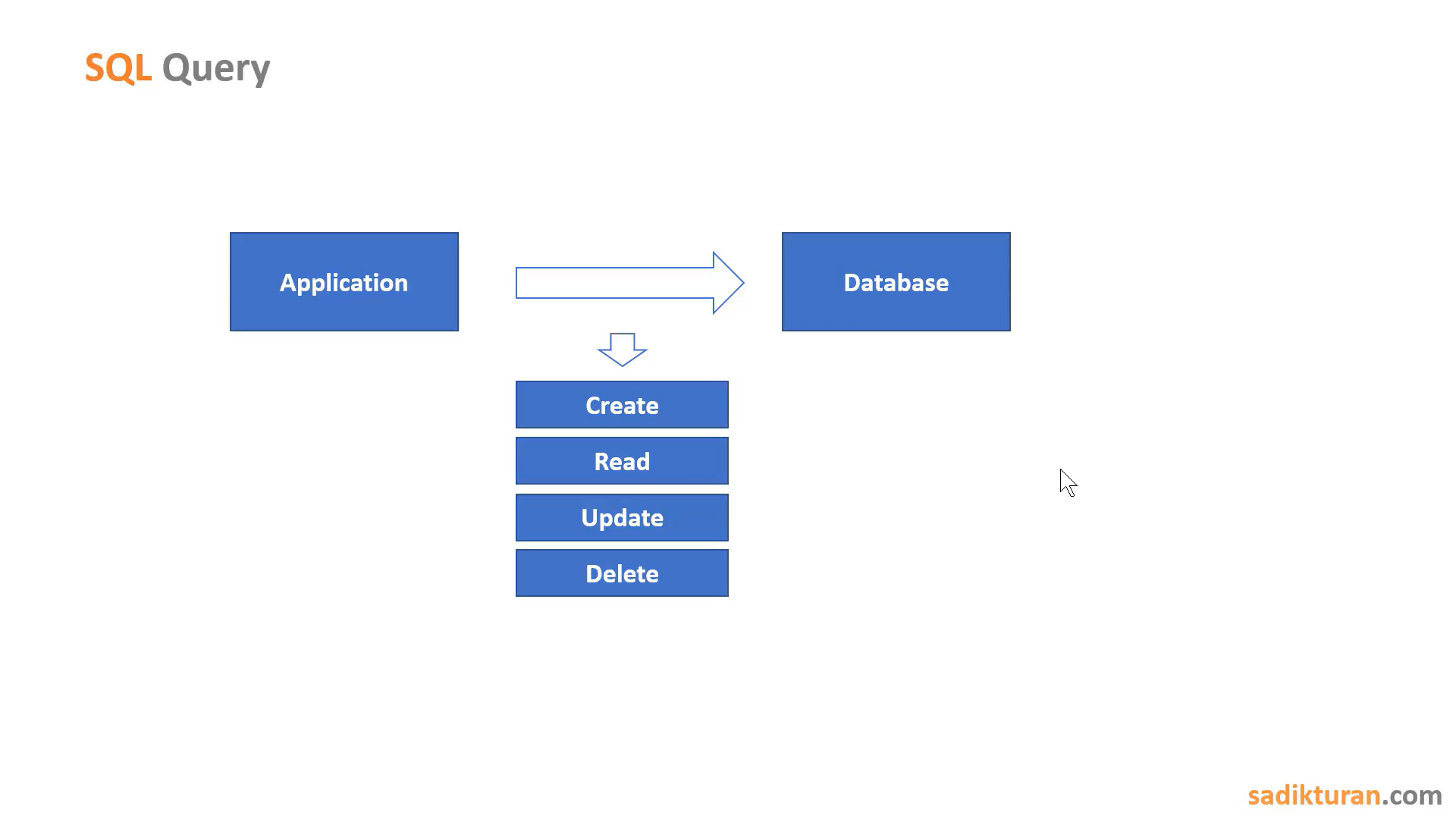
Sql database oluşturuyorsak bu database içerisinde tablolar oluşturmamız gerekir

Örnegin ürün bilgilerini tutmak istiyorsak ürün bilgisiyle ilgili tablo oluşturmamız gerekir. Bu tabloyu excel tablolarına benzetebiliriz.

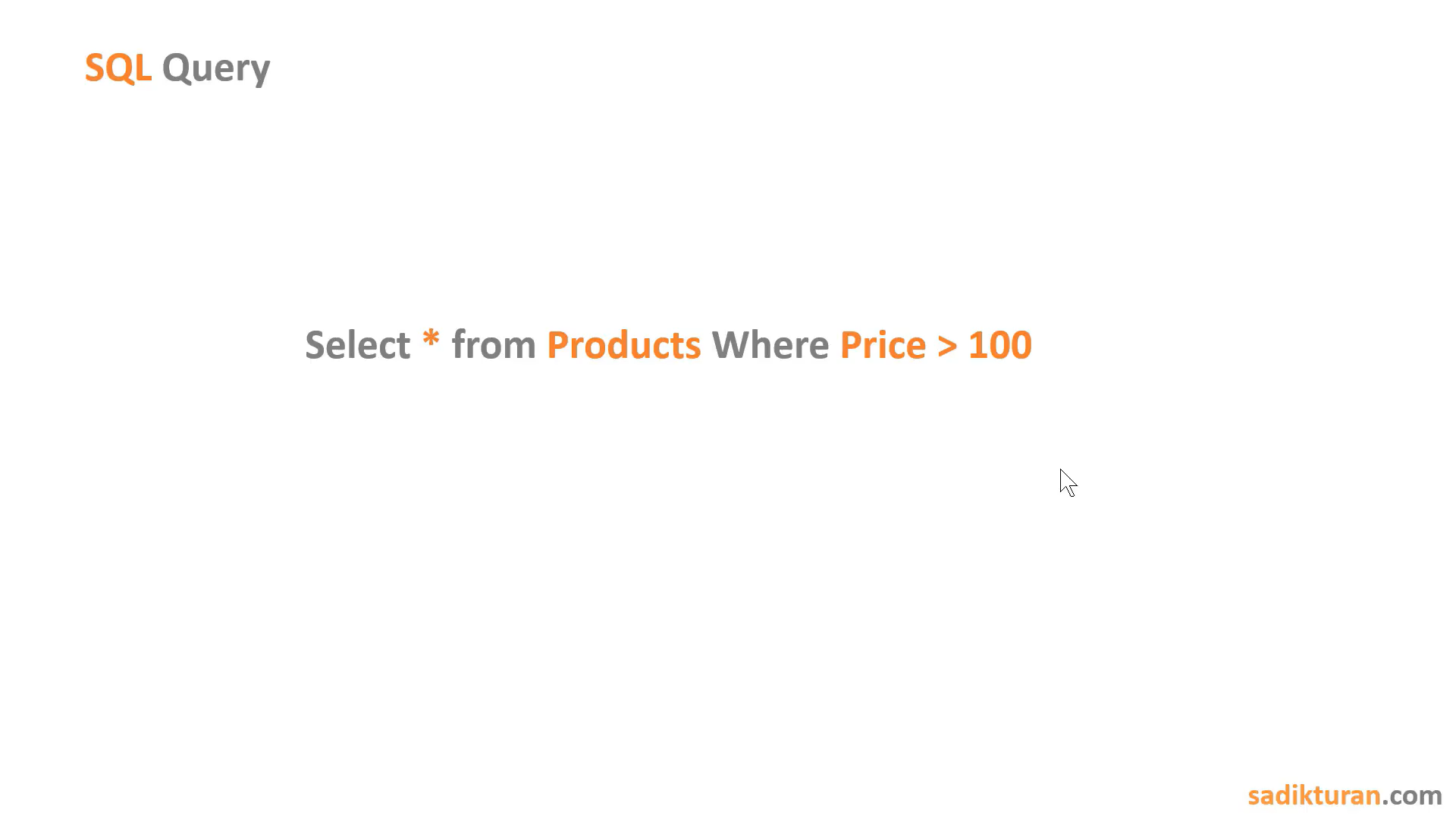




Id ler her zaman primarykeydir

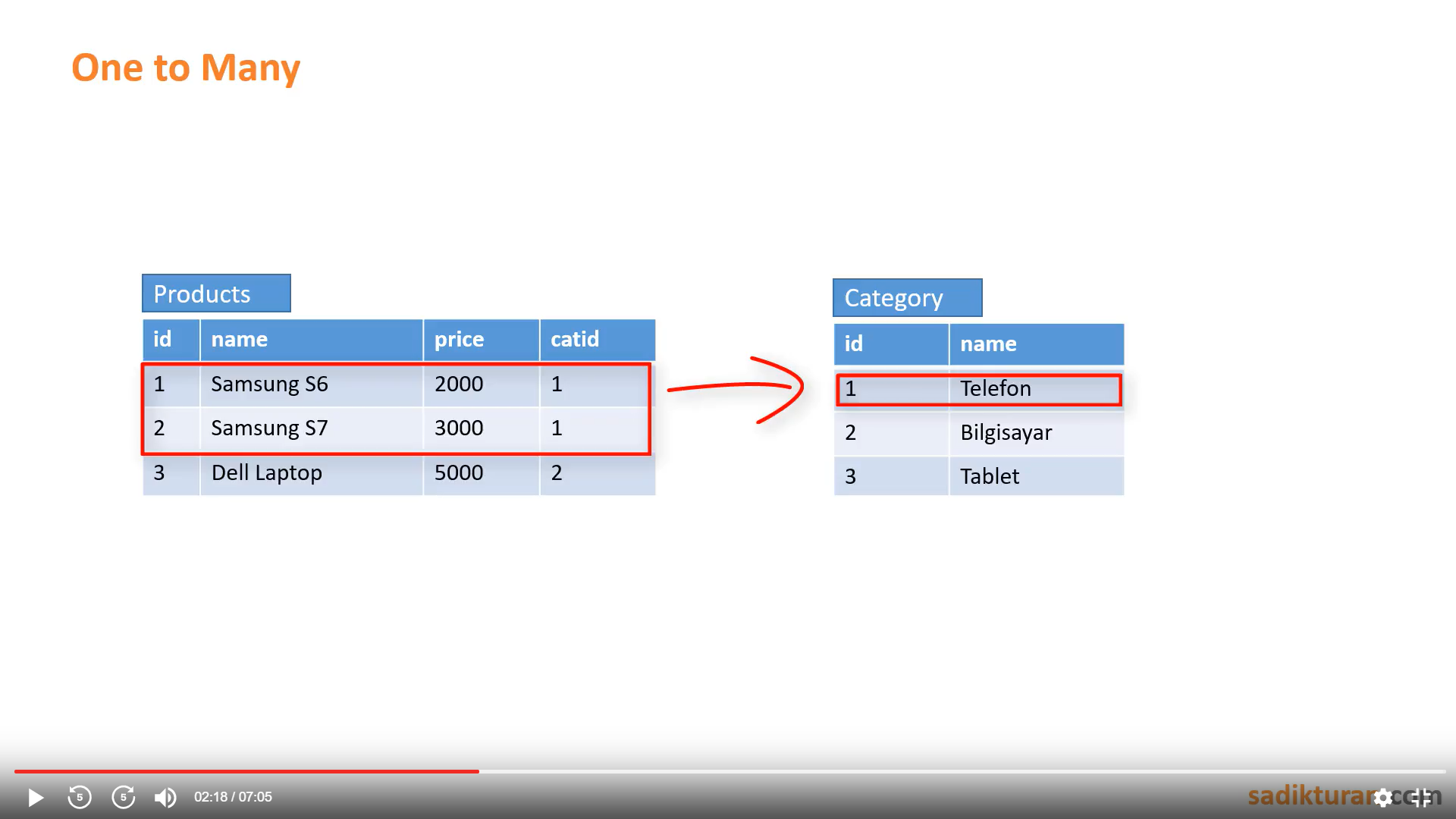


CRUD = CREATE, READ, UPDATE, DELETE



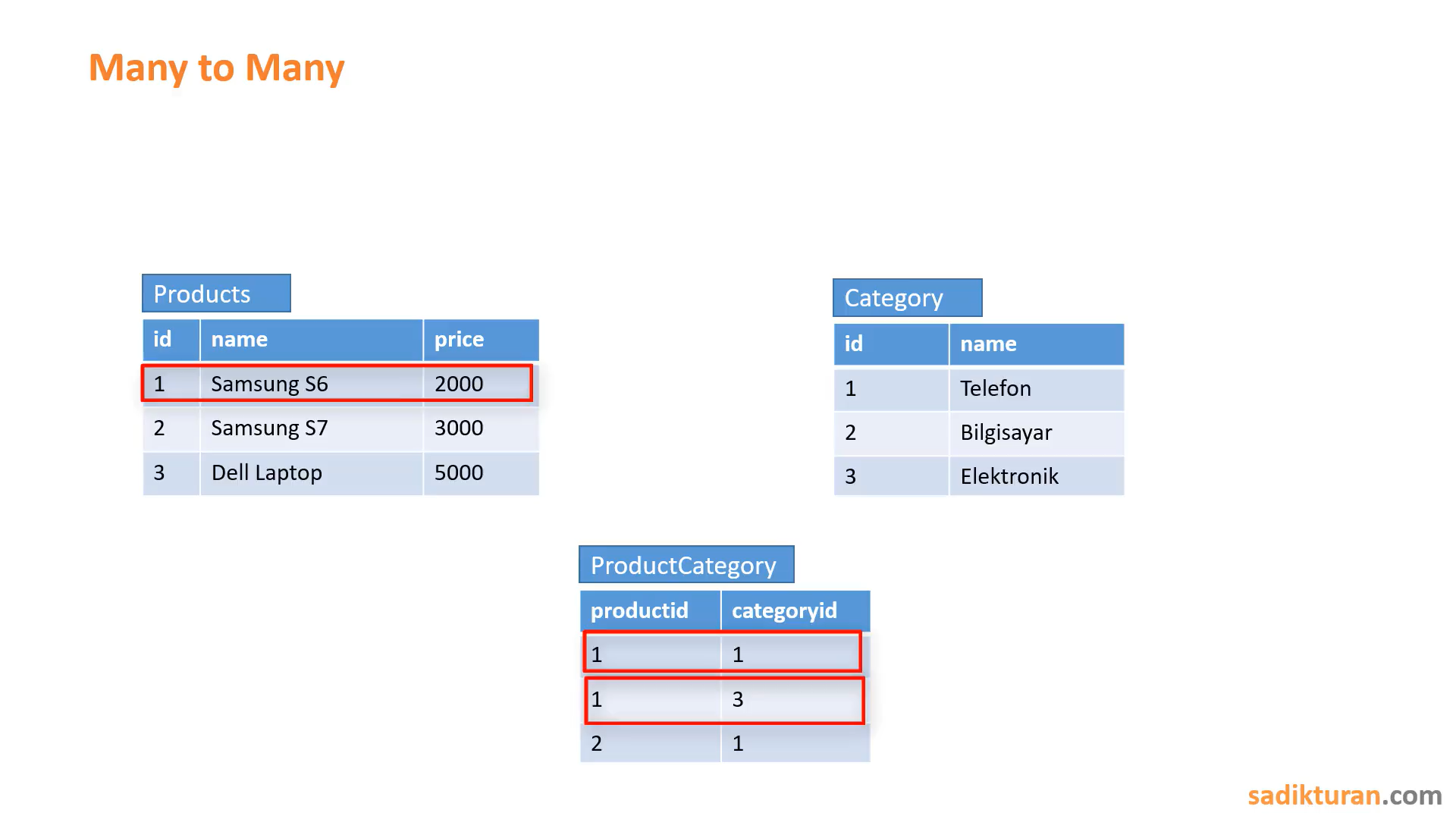
**SQL TABLO İLİŞKİLERİ**

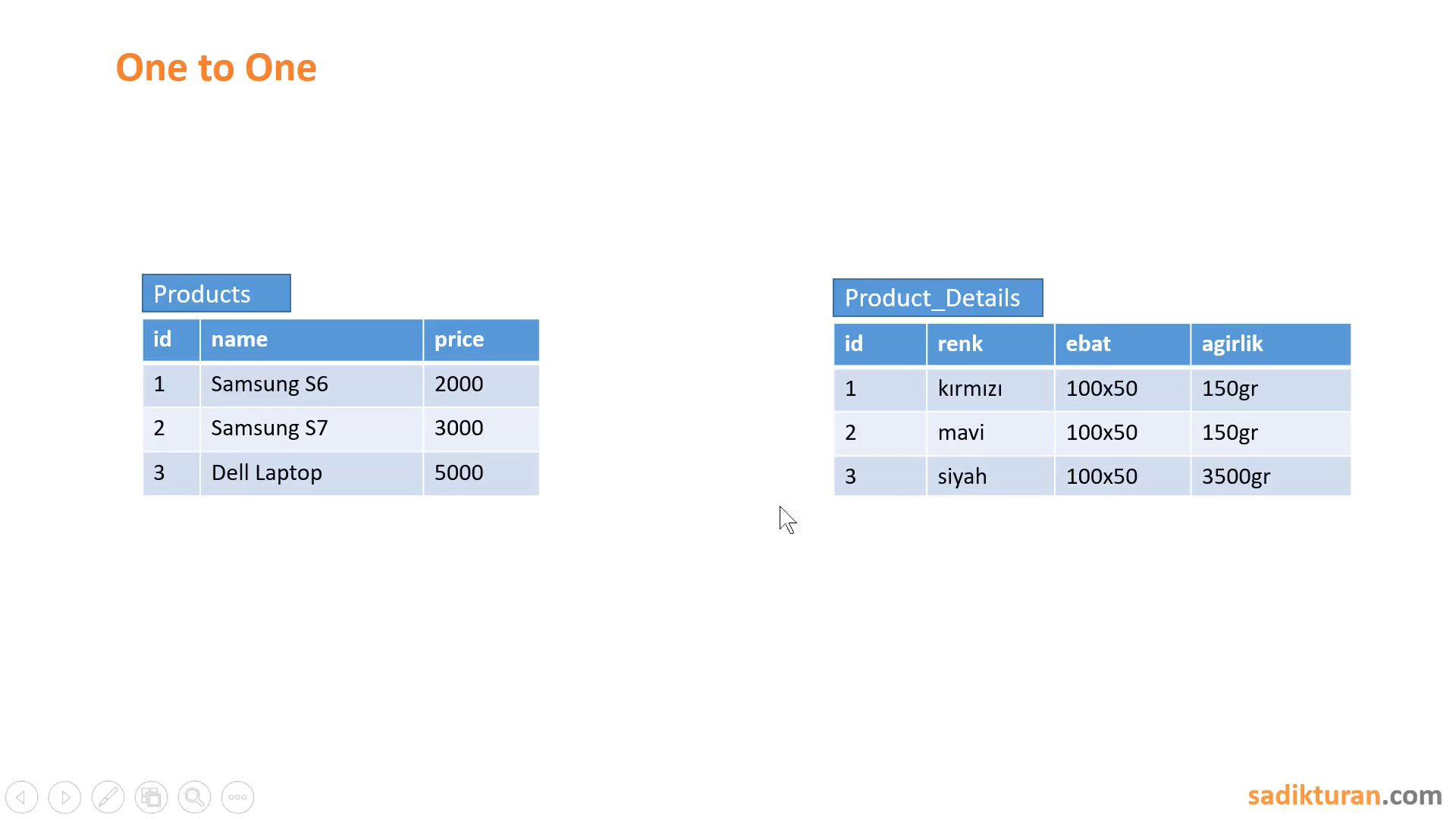




One to many ilişkisinde her ürün 1 kategoriye ait olabilir ancak 1 kategori 1 den fazla ürüne etki edebilir. Değişiklik yapılması durumunda teker teker isim değişmek yerine ayrı bir kategori tablosu oluşturup Id si ile işlem yaparak one-to-many ilişkisi kurmak bizim işimizi kolaylaştırır

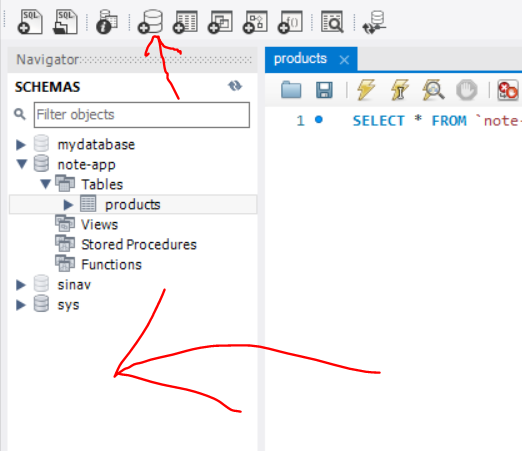
Ancak ben kategori tablosuna elektronik isminde bir kategori eklemek istiyorum ancak ürünler zaten telefon ve bilgisayar yani tablodaki her ürün sonuçta bir elektronik ürün olduğu için ben bunları bu kategoriler haricinde bir elektronik kategorisine de atmak istiyorum yani bir ürüne birden fazla kategori ataması yapmak istiyorum bu durumda many-to-many ilişki tipi kullanmamız gereklidir



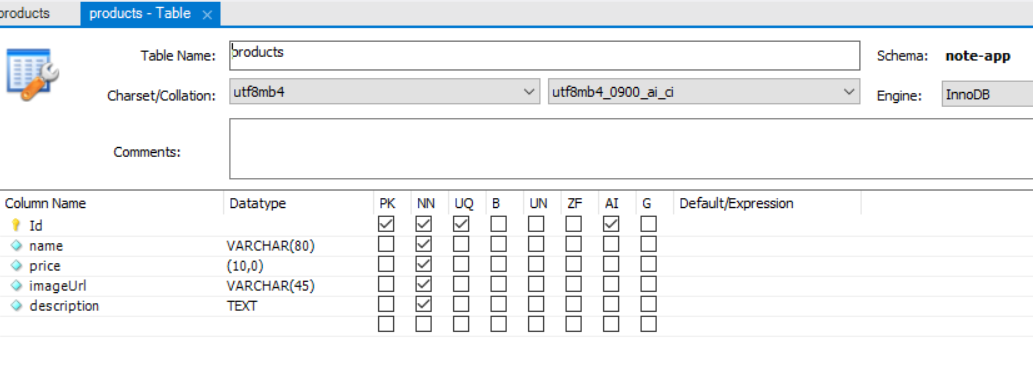


One to one ilişki türünden ürün ve detayındaki tabloda id bilgileri üzerinden detaylandırma işlemi yaparak olası bir değişimde karışıklığın önüne geçmiş oluyoruz. Basit özellikleri products adlı tabloda gösterirken detay tablosunda daha da özelleştirerek işlemlerimizi yapıyoruz

**Veritabanı Tasarımı**



Boş alana sağ tıklayıp create schema yada database simgesine tıklayarak database oluşturabiliriz. Oluşturduktan sonra bizim ilgileneceğimiz kısım ise tablo kısmıdır

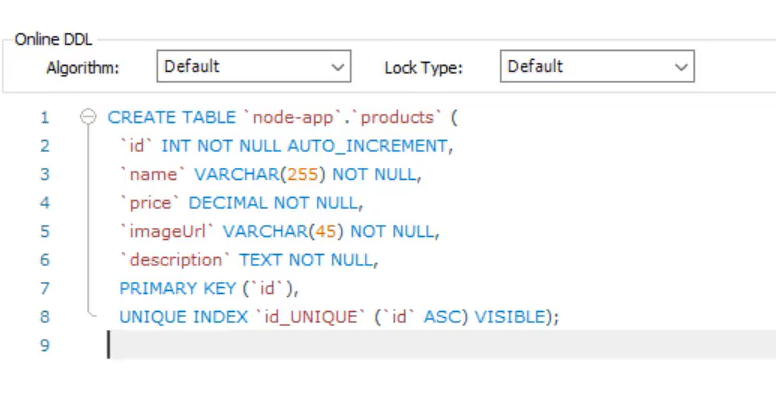


PK = Primary key

NN = Not null

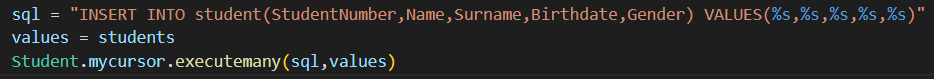
UQ = Unique

AI = Auto Increment



SELECT \* FROM ‘node-app’ .products = node app adlı databaseden product tablosundaki bütün kolonları bana getir

Insert Into sorguları database tablosuna bir kayıt eklerken kullanılır



Veri tabanındaki bir tabloya veri girmek isterken insert into kullanırız

Mycursor.execute(CREATE DATABASE mydatabase“)

Boş bir database oluştururken kullanılan methodtur

Mycursor.execute(CREATE TABLE customers(name(VARCHAR(25)))“)

Tablo oluştururken kullanılan methodtur

Cursor.execute(Select \* From products“)

Products tablosu içerisindeki tüm kolonları seçer

Cursor.fetchall() = execute yaptığımız tablodan birden fazla kayıdı almak isterken fetchall kullanırız

Cursor.fetchone() = execute yaptğımızı tablodan ilk kaydı almak istediğimiz zaman fetchone kullanırız

cursor.execute('Select name,price From Products')

Tüm kolonları seçmek yerine sadece belirli kolonları seçmek istersek kullanırız

Cursor.execute(“select \* from where id=1)

İd si 1 olan kayıt gelir

Cursor.execute(“select \* from where name=’samsungs6’ ”)

Samsungs6 kaydı ile eşleşen kayıdı bize getirir

Cursor.execute(“select \* from where name=’samsungs6’ and price=3000”)

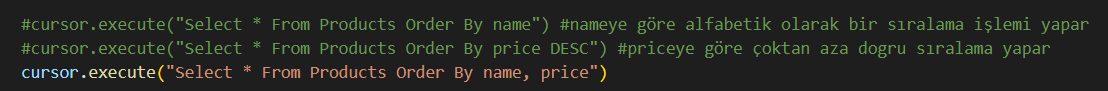
Samsungs6 ve 3000 fiyat kaydı ile eşleşen kayıdı bize getirir



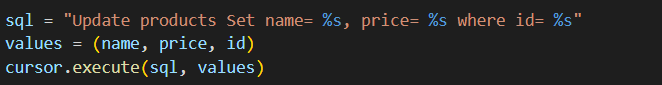
İçinde samsung kelimesi geçen kaydı aramak



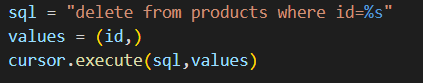
Yüzdeyi başlangıç kısmından kaldırınca başı tam olarak yazdığımız gibi olsun ancak sonu ne olursa olsun arat demek istiyoruz



Order By ile istenen kolonda sıralama işlemi yapılır



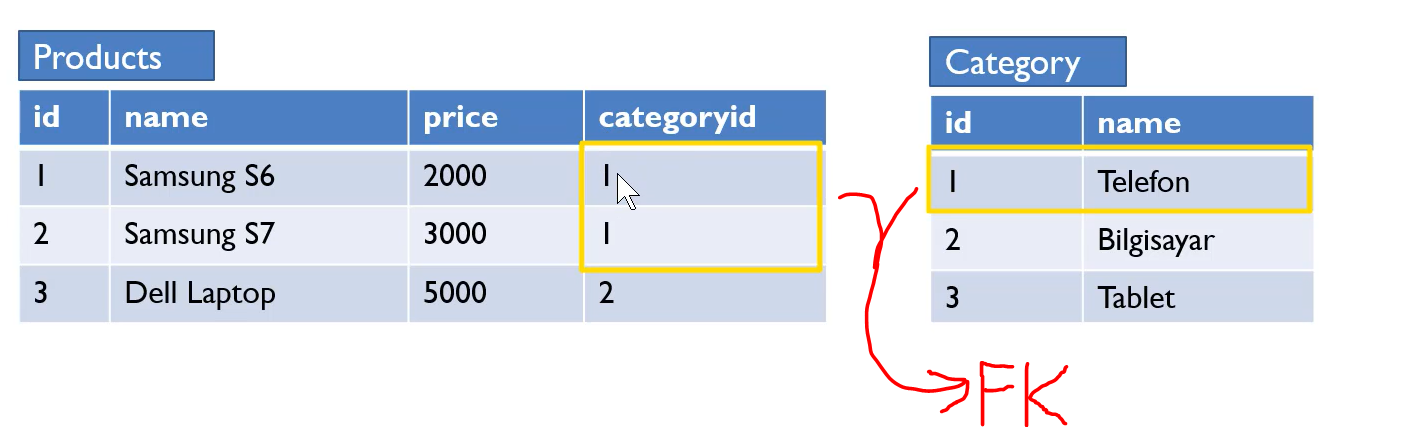
Products tablosunda üzerinde değişiklik yapmak istediğimiz sorguyu seçme işlemidir



Products tablosundan istenilen ıd numarasına göre silme işlemi yaparız



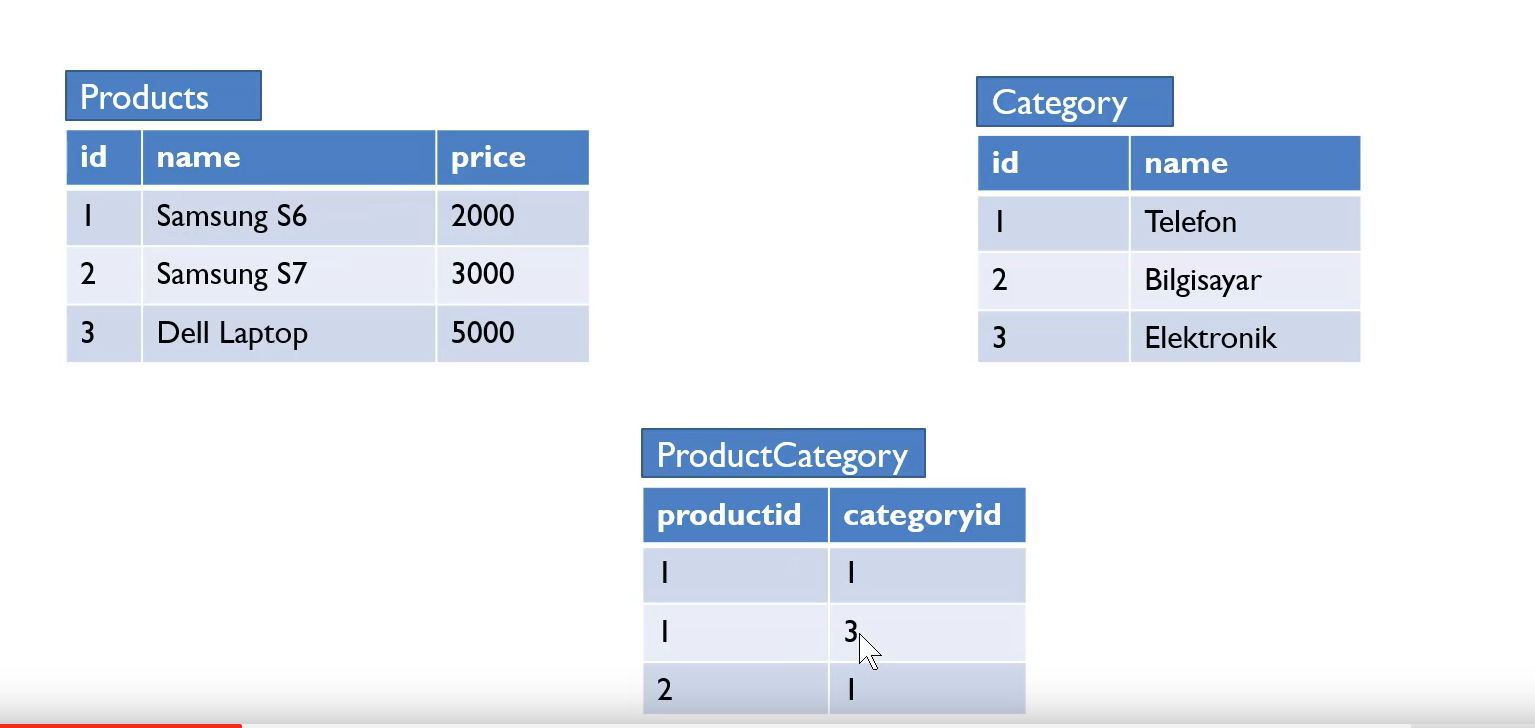
İçerisinde namesi “ s7 ” içeren kaydı sil



**ONE TO MANY FOREIGN KEY FOR CATEGORY**

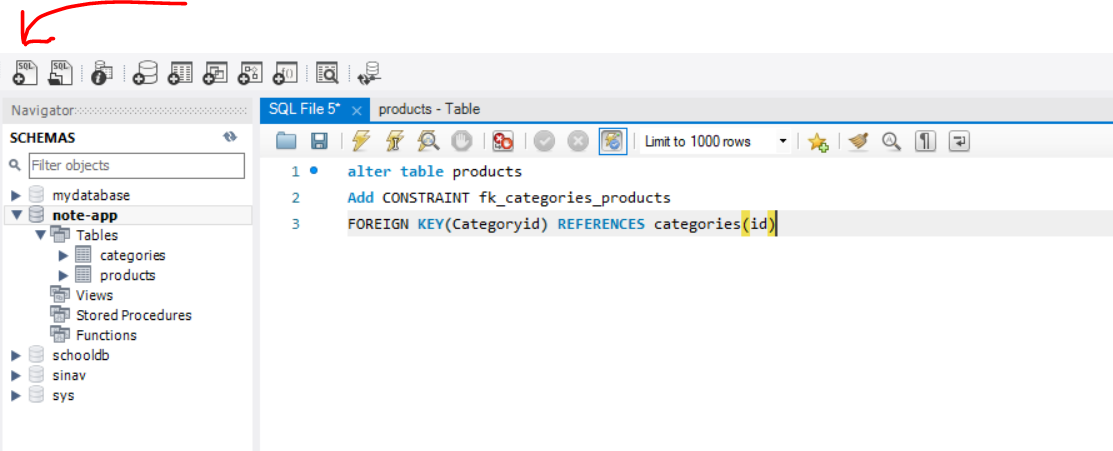
Burada categoryId adlı kolon aslında bir foreign key yani referans tutucudur

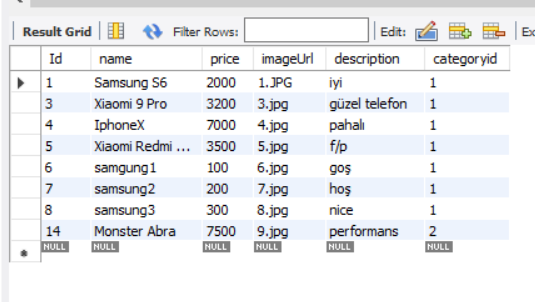
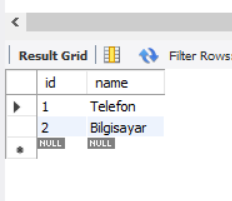
CategoryId kolonu products tablosu için bir foreignkeydir yani bu categoryid nin aslında bir başka tablonun birincil anahtarı olduğudur



**Many to many ilişkisi**

**FOREIGN KEY BAĞLANTISI OLUŞTURMA**



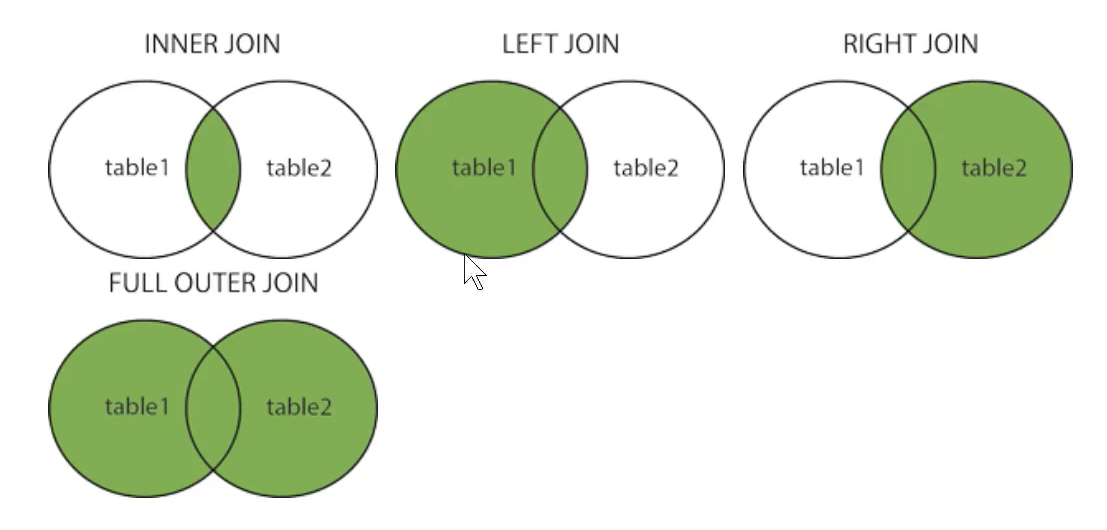
 

Products tablosundaki categoryid ve category tablosundaki id arasında bir foreign key oluşturmak için kırmızı ok ile işaretli sql script tuşuna basarak sorgu yazma ekranını getiriyoruz

alter table products >>>>>>> üzerinde değişiklik yapılacak tablo

Add CONSTRAINT fk\_categories\_products >>>> hangi tablolar arası ilişki kurulacak

FOREIGN KEY(Categoryid) REFERENCES categories(id)>>hangi kolon forreign key olacak ve hangi kolon referans verici olacak bunu belirtiyoruz

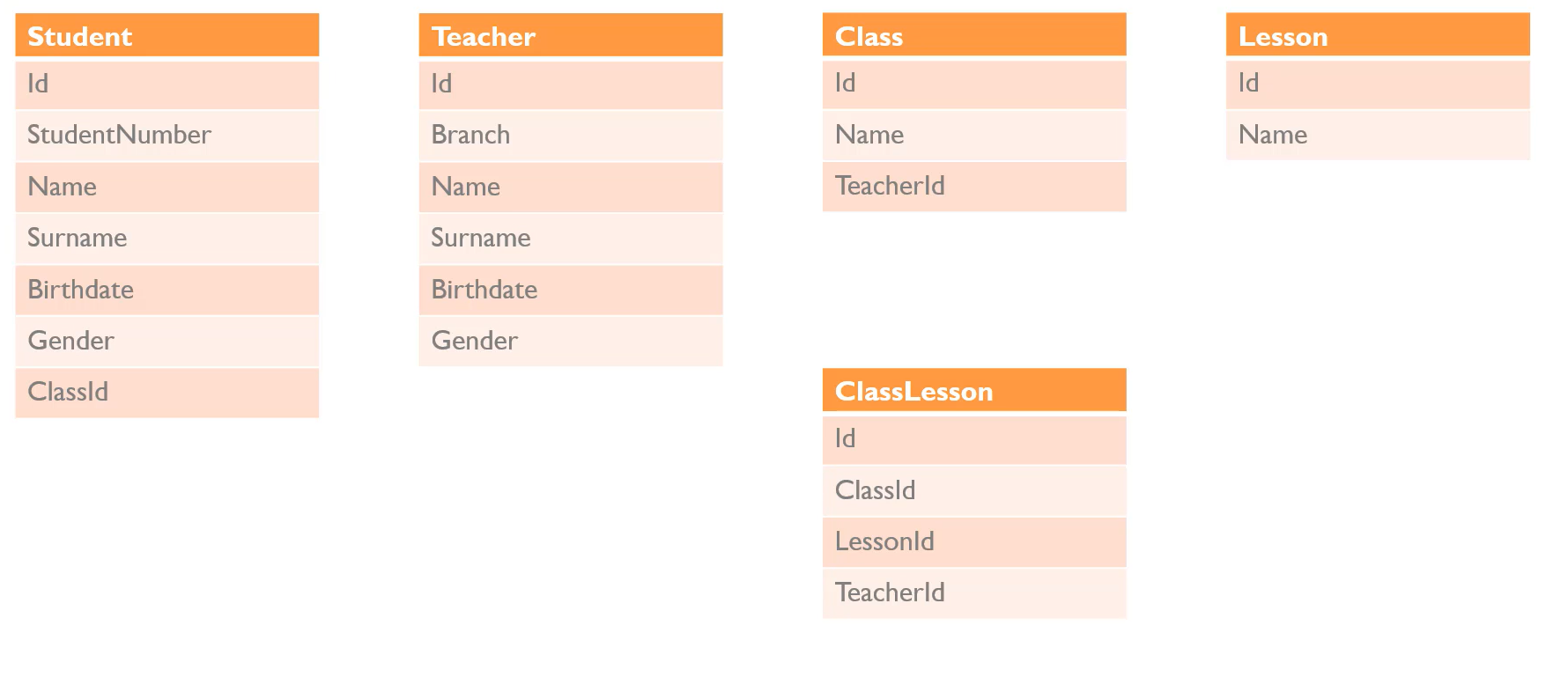


Tablo Özellikleri



Inner join işlemi için categories tablosu üzerinden categories tablosunun id kolonu ile products tablosunun category id kolonu arasında ortak olanları aldık

**ÖĞRENCİ KAYIT PROGRAMI VERİTABANI TASARIMI**



Mysql yada sqlite veritabanı sql database grubu altında ele alınan veritabanı türleridir

-Mysql bir server tabanlı veritabanıdır yani mysql databasesini kullanabilmek için yerel bilgisayarımıza bir server kurduk. Yarın öbür gün uygulamamızı yapıp yayınladığımız zaman uygulamamızın kullandığı databaseyi de yayınlamış olmamız gerekiyor bir hosting firmasından mysql server satın alıp bu servere databasemizi yüklememiz gerekir

-Sadece bir uygulamaya özel bir databaseye ihtiyaç duyuyorsak bu durumda sqlite veritabanını kullanabiliriz.

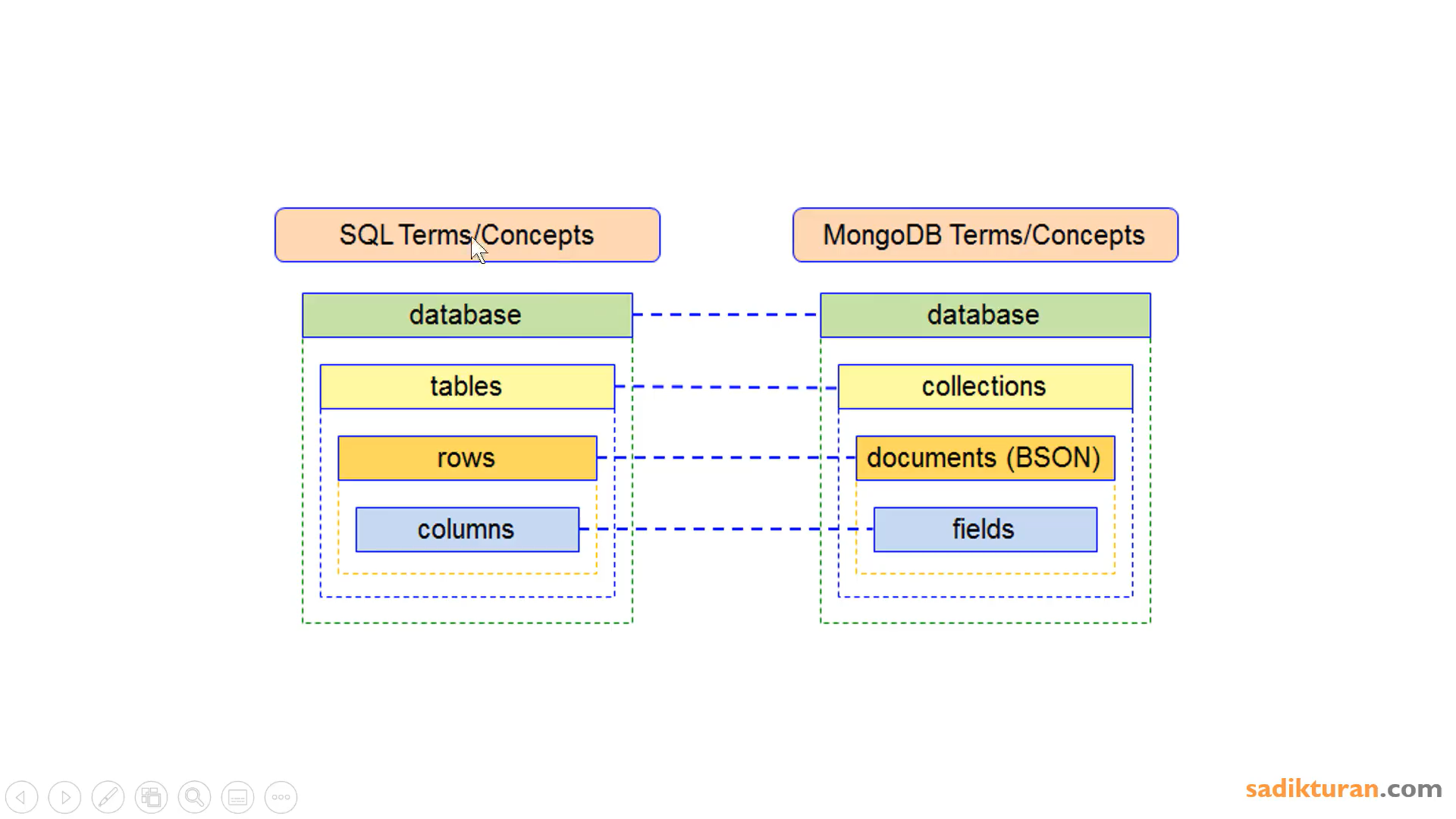
-Örneğin bir oyun yaptık ve bu oyun sadece o mobil uygulamayı yükleyen kişinin oynadığı bir oyun dolayısıyla o uygulamaya ait verileri sqlite veritabanında kullanabiliriz server tabanlı bir databaseye ihtiyacımız yok

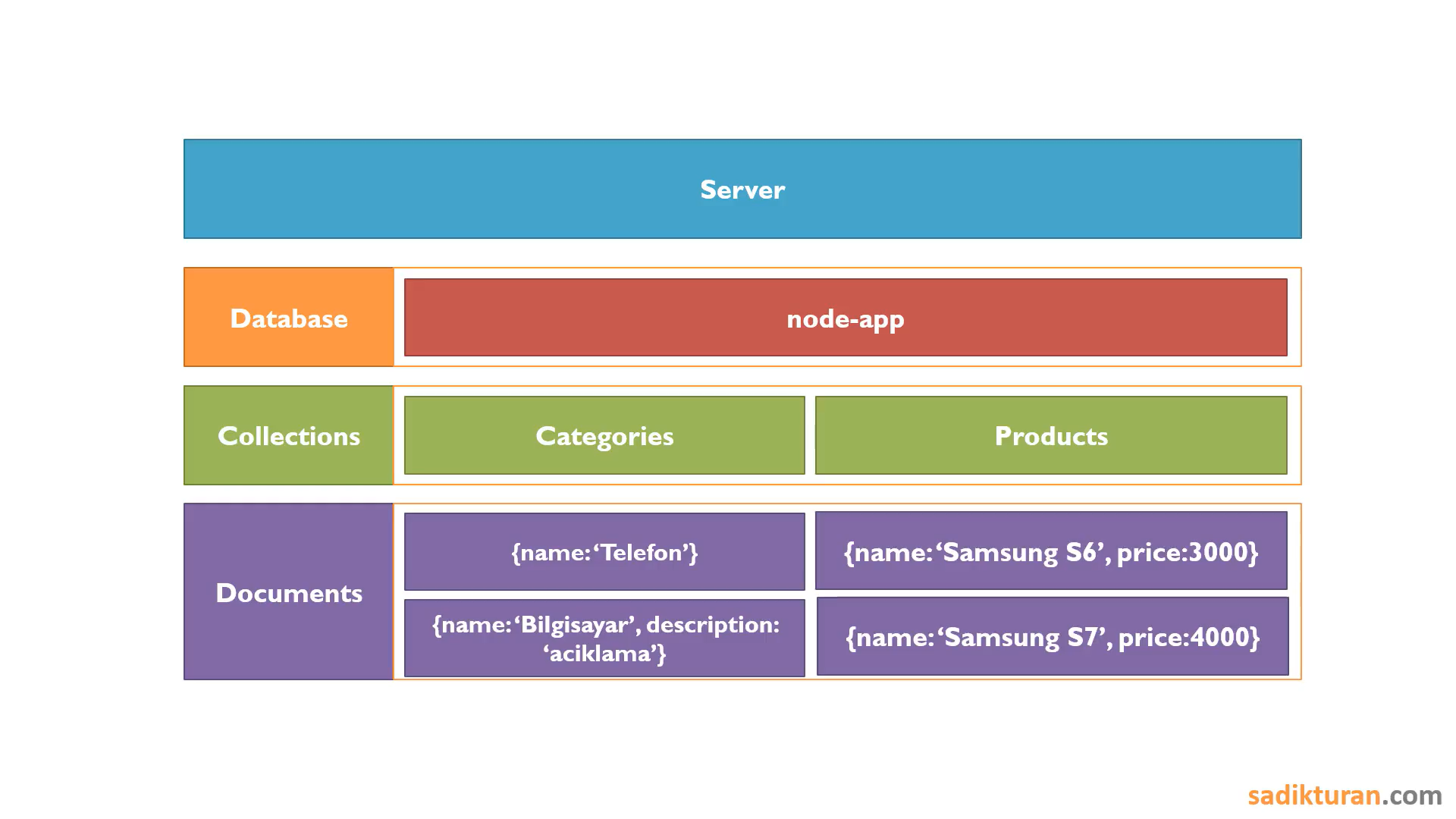
-Eğer ki oyun birçok kişi tarafından çevrimiçi oynanacaksa ve aynı anda aynı datalara birden fazla kişi ulaşmak istediği zaman bir server tabanlı database kullanmak zorundayız

-Bir json, txt yada excel dosyası oluşturup verileri buraya yüklemek yerine sql kullanmamızın sebebi sql üzerinde sorgularla işlemlerimizi kolayca gerçekleştirebilmemizdir

**NOSQL**

Nosql yapısında tablolar, satırlar, kolonlar yoktur. Bunlar yerine bir json datayı nosql database içerisinde herhangi bir bağımlılık olmadan kullanacağız. Mesela ürünün adı ve fiyatını bir kayıtta yolladık ancak diğer kayıtta istersek yollamayabiliriz, esnek bir yapı söz konusudur. Nosql grubu altında en popüler olan veritabanı MongoDb dir ve mongodb tıpkı mysql gibi server tabanlı bir veritabanıdır.







Artan sekilde sıralama 1, azalan sekilde sıralama -1 dir.