

1. JSON "fetch" metodu
 - a. fetch metodu JSON veri kaynağına ulaşmayı sağlayan metod
 - b. async fonksiyonlar asimetrik(eş zamansız) bağlantı sağlar
 - c. await kelimesi asimetrik fonksiyonlarda diğer işlemlerin bloke olmadan beklemede kalmasını sağlar.
 - d. try-catch kontrol bloğunda try istenen işlemi dener catch hatayı yakalar.
2. JSON Rest API bağlantısı
 - a. GET metodu veri kaynağından verileri elde etmek için kullanılır.
 - b. POST metodu veri havuzuna ekleme yapmak için kullanılır.
 - c. Content-type: API içeriğinin JSON tipinde olduğunu
 - d. Authorization: API kaynağına erişim yetkisi için key(anahtar) tanımı
3. JSON Server

```
"npm init" komutu ile package.json ayar dosyasını oluşturur.  
b. "npm install --save json-server" komutu ile json-server ayarlarımızı kaydeder.  
c. package.json içerisinde "test" kısmı yerine "json:server" isimli bir key alanı oluşturup value kısmına "json-server --watch db.json" yazalım.  
d. "npm run json:server" yazarak lokal sunucu üzerinde json-serveri çalıştıralım.
```

4. İlişkisel Veritabanı
 - a. İlişkisel veritabanlı düzenli ve sistematiktir.
 - b. Örnek veritabanı yönetim sistemleri SqlServer,MySQL
 - c. Tabloların ilişkisel olup olmadığını bir tablodaki veri silindiğinde diğer tablodaki bağlı kayıtların da silinmesi gerekiyor mu sorusu sorularak bulunabilir.
5. İlişkisel Veritabanı Tablolar
 - a. Tablolardaki veri tipleri temel olarak sayısal ve sözel şekilde ayrılır.
 - b. int tipi tamsayı için, text tipi metinsel veriler için kullanılır.
 - c. date tipi tarihler,datetime tarih-saat,money para birimi,image dosya tipi
 - d. image tipinde veri saklamak veritabanının şişmesine ve yavaşlamasına neden olur. Bu nedenle resmin kendisi yerine adres bilgisini saklamak daha doğrudur.
6. İlişkisel Veritabanı Anahtarlar
 - a. Birincil Anahtar tabloda kayıtlı her bir veriyi benzersiz kılar.
 - b. Yabancı Anahtar ilişkili tabloya bağlanmak için kullanılır.
 - c. Birincil Anahtar ve yabancı anahtarın tipi aynı olmalıdır.
 - d. Sql Server'da birincil anahtara otomatik sayı ataması için "Identity Spesification" ayarı altından "Is Identity→Yes", "Identity Seed" başlangıç değerini, "Identity Increment" artış miktarını belirler.
7. Sql Komutları
 - a. INSERT – Ekleme
 - b. DELETE – Silme
 - c. UPDATE – Güncelleme
 - d. SELECT – Seçme/Listeleme(SELECT * FROM Sehirler WHERE SehirBolge=104)
8. Entity Framework Tanımı
 - a. Ado.Net veritabanı kütüphaneleri içerisinde yer alır.
 - b. Veritabanında yer alan tablo ve alanları nesne olarak kullanmamıza imkan veren bir yazılım mimarisidir.
9. Entity Framework Bağlantısı

```
1 namespace CografyaDbEntityFramework.Models  
2 {  
3     public class Bolge  
4     {  
5         public int BolgeNo { get; set; }  
6         public string BolgeAdi { get; set; }  
7         // Diğer özellikler  
8     }  
9 }
```

- a.
 - b. UseSqlServer metodu veritabanı yönlendirilecek veritabanı sunucusun "ConnectionString" yani bağlantı cümlesini belirler.
- ```
options.UseSqlServer("Server=DESKTOP-2B368LT\\SQLEXPRESS;Database=CografyaDb
;Trusted_Connection=True;");
```
10. NoSql Veritabanı
    - a. Google Firebase yapısı altındaki "Firestore" bulut tabanlı ilişkisel olmayan JSON(text) formatında bir veritabanı yönetim sistemidir.
    - b. Firebase kütüphanesini uygulamaya çağırma: `import { initializeApp } from "firebase/app";`
    - c. Çağırdığımız kütüphaneden firebase yükleme metodu `const app = initializeApp(firebaseConfig);`