

## Senaryo:

Okul veritabanıyla entegre çalışan bir serviste, **her öğrencinin aldığı dersler ve bu derslerden elde ettiği notlar paylaşılmaktadır.**

İncelediğiniz kod bloğunda, “önceki yazılımcının” 3 adet iç içe döngü kullandığını fark ediyorsunuz.

Aylar sonra servis performansının ciddi şekilde yavaşladığı raporlanıyor ve bu durumu iyileştirmeniz talep ediliyor.

Kodu incelediğinizde aşağıdaki durumlarla karşılaşıyorsunuz:

**Birinci Döngü:** Tüm öğrenciler üzerinde dolaşılır.

**İkinci Döngü:** Her öğrencinin aldığı dersler üzerinde dolaşılır.

**Üçüncü Döngü:** Her ders için öğrencinin aldığı sınav notları üzerinde dolaşılır ve bu notlar ekrana yazdırılır.

## Sorular:

1. Mevcut durumunun zaman karmaşıklığı nedir?
2. Kodu okuduğunuzda 3 tane iç içe döngü var ne her döngü başlangıcında veritabanından sorgu yapılmış. Okunabilirliği çok düşük. Tüm bu anlatılanlar düşünüldüğünde ve modern teknikleri de kullanarak nasıl bir iyileştirme yapılır?
3. Student, Course ve Grade tablo isimleri ile örnek bir veritabanı tasarımını yapar mısınız?

**Student tablosu** temelde minimum şu alanları içerir: first name, last name ve created date.

**Courses tablosu** temelde minimum şu alanları içerir: course name, course code.

**Grade tablosu** temelde minimum şu alanları içerir: exam score, exam date.

Not: Tablolara anahtar atanabilir, ilişki kurmak için ek anahtar verebilirsiniz.

- a. 2019 yılında kayıt olan ve **MAT101** dersini almış olan tüm öğrencilerin **adını, soyadını ve sınav puanlarını** listeleyen bir SQL sorgusu yazınız.
- b. Her bir ders için, o derse ait en yüksek **sınav puanını** ve o puanı alan öğrencilerin **adını ve soyadını** listeleyen bir SQL sorgusu yazınız.
- c. Öğrencilerin aldığı derslerden en düşük **sınav puanını** alan öğrenci ve dersi listeleyen bir SQL sorgusu yazınız.