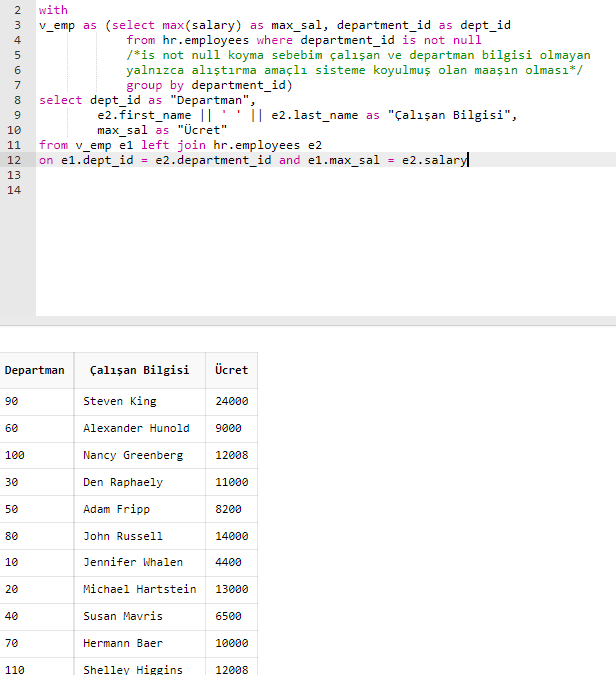
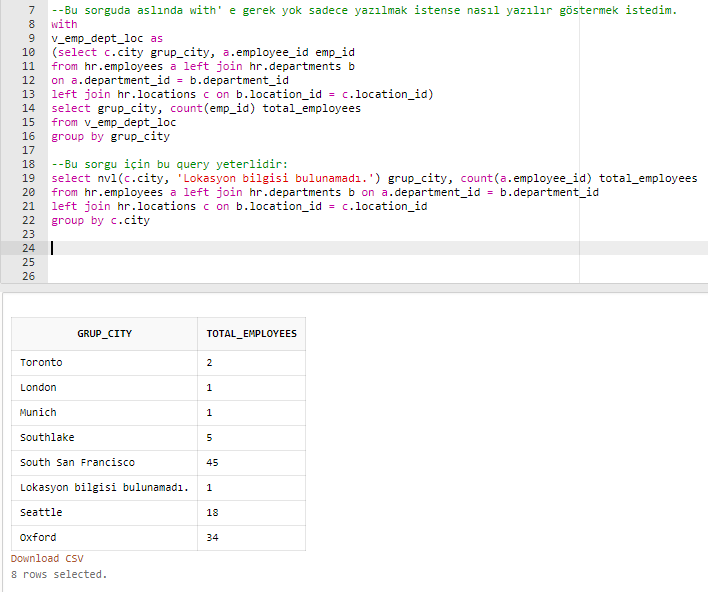
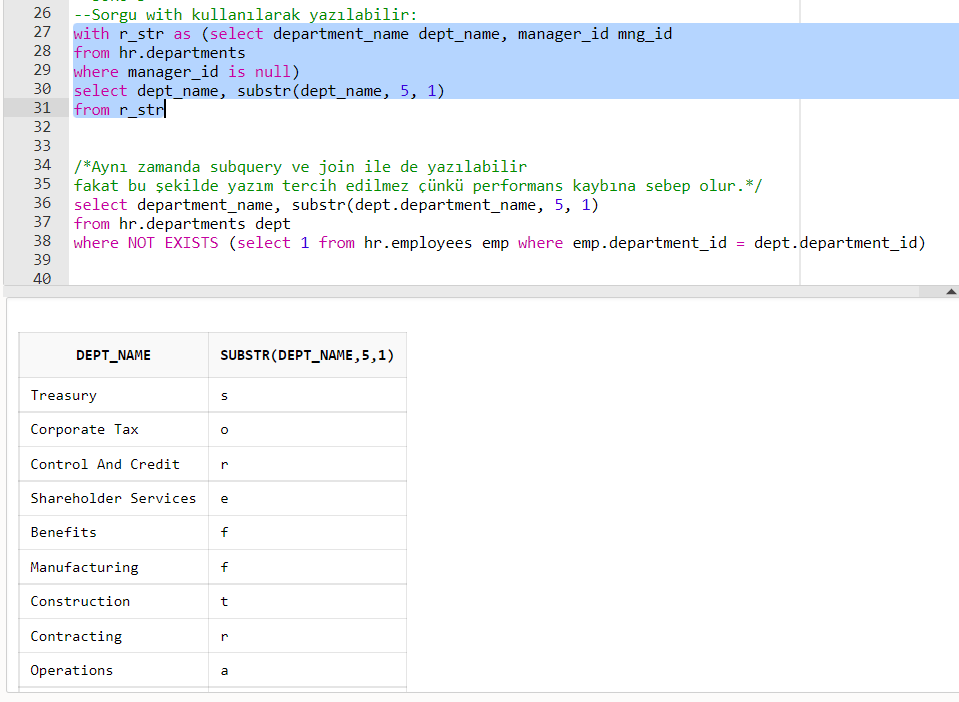
1. Return all employees who earns the highest salary in each department. (Hint: use analytic functions)



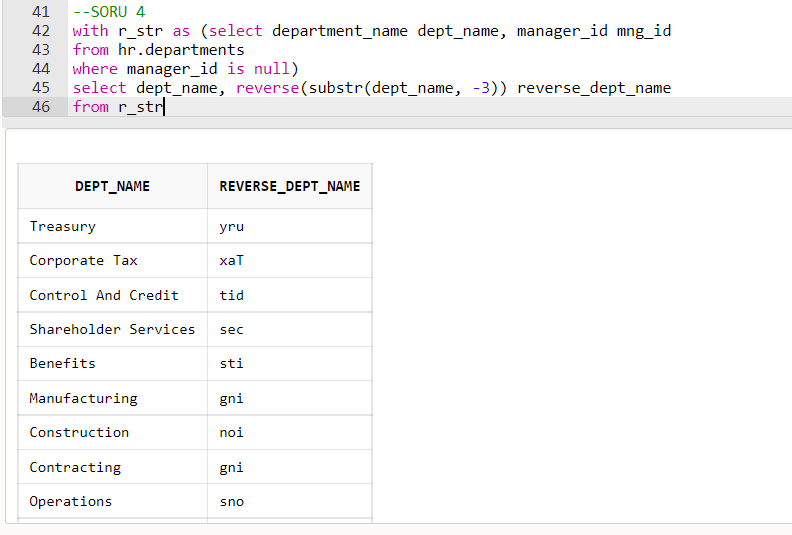
1. Return number of employees registered for each location.



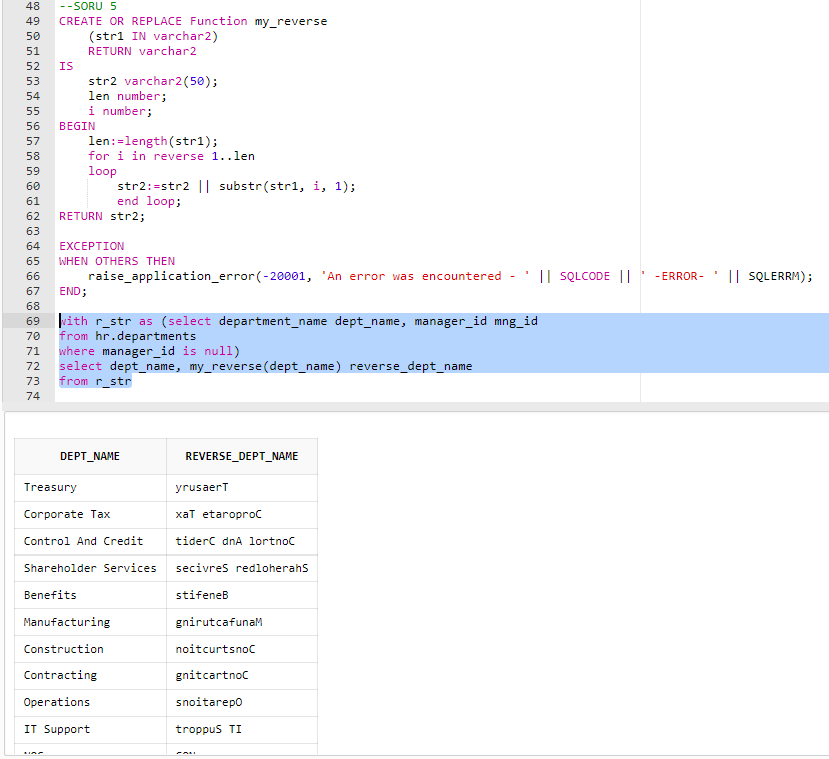
1. Some departments don’t have any registered employee record. Return 5th letter of those department names. (Don’t try to understand logic here because there is none. It’s just practice.)



1. (Question 3 cont.) Return the string between 1st letter from the end and 3th letter from the end. . (For example => input = ‘Benefits’, output = ‘sti’).



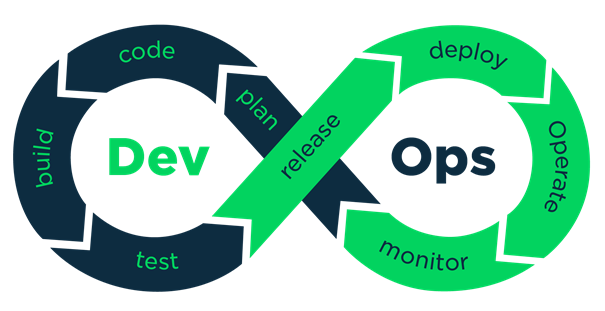
1. (Question 3 cont.) Return reverse of those department names. (For example => input = ‘Benefits’, output = ‘stifeneB’). (HINT: do not use reverse function)



1. **DevOps**

**Neden ihtiyaç duyulur ?**

DevOps’ a neden ihtiyaç duyulduğunu bir örnekle anlatmak gerekirse; yazılım üretiyorsanız veya yazılım kullanıyorsanız günün sonunda bir takımınız, bir yazılımcı ekibiniz vardır. Yazılım takımı, yazılım geliştirir veya yazılım üretir. Bir tarafta da başka bir takım var ve bu yazılımı alıp sunuculara yerleştiriyor, deployment’ ı yapıyor, ayağa kaldırıyor ve o sunucuların hayatta olmasından, sağlıklı olmasından sorumludur. Bu iki takım arasında problemler yaşanabilir. Örneğin deployment’ ı yapan takım yazılımı sunucuya koyduklarında çalışmadığını veya koyarken problem çıktığını bu sebeple rollback yaptıklarını, bir öncekinin çalıştığını söyleyebilir. Yazılımcılara gidilip sorulduğunda kendi makinalarında çalıştığını, sorunun deployment’ ı yapan ekipte olduğunu ifade edebilirler. Bu şekilde iki ekip arasında böyle problemler yaşanabilir. Bunu önlemek için bütün bu süreçleri baştan aşağı tanımlamak, dokümante etmek gerekebilir. Yazılımcılar deployment nasıl yapılıcağını adım adım anlatmak durumunda kalabilirler. Ama buradaki sorun o doküman hazırlanırken her şeyin iyi gideceği düşünülerek yazılmasıdır. Deployment’ ı yapan ekip bu dokümanı takip ederek dahi problemler yaşayabilirler. Bu dokümanın her seferinde iyileştirilerek daha iyi bir hale getirilmesi de uzun sürecektir. Tam olarak bu noktada önemli olan bu takımların beraber çalışması gerekliliğidir. Bu ihtiyaçtan kaynaklı olarak DevOps kültürü oluşmuştur. DevOps kavramı ile beraber takımların birbirleriyle yaptığı çalışmalar ve iletişim yoğunluğu sayesinde devreye alınması gereken yenilik ve düzeltmeler çok kısa süre içinde işleme alındıkları için verim ve başarı yüksek olmaktadır. DevOps, son kullanıcılara sürekli değer teslimi sağlamak için insanlar, süreç ve ürünlerin birliğidir. DevOps tanımına göre ilk yapılması gereken insanları birleştirmektir, tek bir hedef etrafında olduklarının bilincini o insanlara verebilmektir. Şirket içi kültür olarak müşterilere değer aktarılması gerektiğini ve bunu engelleyen her şeyin aslında şirket içerisindeki herkesin problemi olduğunu ve bunun çözülmesini gerektiğini anlatabildiğiniz ve bunu bir kültür haline getirebildiğiniz anda ciddi bir problem çözülmüş olunur. Bir sonraki aşama süreçlerdir. Nasıl tanımlayacağımız ve nasıl yapacağımız sorularının cevaplarının bütünüdür. Son olarak ürünler gelir çünkü hangi ürünü kullandığınız önemli değildir. Bunun sebebi ilk başta DevOps kültürü yaratılamadıysa hangi ürün seçilirse seçilsin başarılı olma ihtimali düşük olur.



DevOps, Bilgi Teknolojileri departmanı içerisinde bulunan iki temel birimi (Developers and Operations) Geliştiriciler (Yazılım Geliştiriciler, Yazılım Testçileri, vb.), Operasyon (Sistem Mimari ve Altyapı Ekipleri, Güvenlik ve Ağ ekipleri vb.) bir arada etkili bir iletişim içerisinde beraber çalışmalarıdır.