1- SAP Commerce (Hybris) nedir? Hangi amaçlarla kullanılır? Kullandığı teknolojiler nelerdir? Kısaca açıklayınız.

SAP Hybris, şirketlerin dijital ticaret operasyonlarını yönetmelerine yardımcı olur. Hybris tanımını basit bir şekilde yapmak istersek tam anlamıyla bir e-ticaret platformudur. İşletmelere çok kanallı satış, müşteri ilişkileri yönetimi, ürün yönetimi, sipariş yönetimi ve pazarlama gibi bir dizi özellik sunar. SAP Commerce, büyük ölçekli şirketler ile entegre olarak esnek ve kişiselleştirilebilir bir çözüm sunar.

SAP Commerce, Java ve Spring Framework teknolojilerini kullanır. Veritabanı olarak genellikle Oracle veya Microsoft SQL Server gibi ilişkisel veritabanları kullanır. Bunların yanında RESTful API'ler ve SOAP tabanlı web servisleri gibi standartlar aracılığıyla dış sistemlerle iletişim kurar.

2- Birbirinden bağımsız iki platformun birbiriyle haberleşmesi nasıl sağlanabilir? Örneğin, X platformu Java ile yazılmış olsun, Y platformu C# ile. Bu iki platform bir biri ile iletişim halinde request-response ilişkisi kurması gerekiyor. Bu yapıyı nasıl sağlarız? Bu iletişim sırasında güvenlik nasıl sağlanır?

Birbirinden bağımsız iki platformun iletişimin sağlamak için birkaç farklı yöntem bulunmaktadır. Bunlardan bazıları: API'ler, SOAP ve mesaj kuyruklarıdır. Bu iletişim esnasında güvenli bir iletişim kanalı kurmak için HTTPS protokolünü kullanabiliriz. Bunun yanı sıra web tabanlı kimlik doğrulaması için JWT kullanabiliriz.

3- SQLR nedir? Kullanım alanlarını araştırınız. Kurumsal bir projede kullanabilecek iki farklı kullanım alanı örneği veriniz.

Apache Software tarafından geliştirilen açık kaynaklı bir arama platformudur. Java ile yazılmıştır. Programcıların gelişim süreçlerindeki işlerini kolaylaştırma amacı güderek gelişmiş arama kriterleri ile yüksek hızda veri sunar. SOLR'ın özellikleri arasında metin analizi, indeksleme, sorgulama, yüksek kullanılabilirlik, dağıtık arama, dinamik clustering ve XML/JSON gibi farklı veri formatlarını destekleme gibi özellikler bulunur. Web arama, e-ticaret, büyük miktarlarda içeriklerin indekslenmesi, büyük veri kümelerinin analizi gibi alanlarda kullanılır.

Örnek:

1- E-ticaret optimizasyonu: Kullanıcıların daha hızlı ve doğru bir şekilde ürünleri bulmasını sağlamak için kullanılabilir
2- Big-Data analizi: Büyük verilerini analizi gerçekleştirmeyi hedefleyen bir kurum SOLR kullanabilir. Çeşitli kaynaklardan gelen büyük veri kümelerini indekslemek, analiz etmek ve kullanıcıların bu verilere erişimini sağlamak için kullanılabilir.

4. soru açıklamalarını aşağı ekledim

SORU 4:

1- Java'da 100 adet random sayıya sahip bir liste oluşturun.

```
public static List<Integer> getRandomArrayList(int size, int bound) {
    List<Integer> list = new ArrayList<>();
    for (int i = 0; i < size; i++) {
        list.add(getRandomNumberBetweenZeroTo(bound));
    }
    return list;
}
public static int getRandomNumberBetweenZeroTo(int upperBound) {
    // Sıfır ile üst sınır arasında sayı üretmek için, üst sınırı bir artırıp random() metodu
ile çarparız
    return (int) (Math.random() * (upperBound + 1));
}
```

2- Daha sonra bu listenin bir kopyasını oluşturun.

```
List<Integer> list = getRandomArrayList(100, 100);
List<Integer> copyList = new ArrayList<>(list);
```

3- 0 ile 100 arasında bir sayı üretin.

```
// IndexOutOfBound istisnasını almamak için 0 ile 99 arasında rastgele sayı oluşturuyoruz
int randomIndex = getRandomNumberBetweenZeroTo(99);
```

4- Kopya listedeki bu random sayının olduğu indeksteki değeri silin.

```
System.out.format("Silinecek olan sayı: [%d]: %d\n", randomIndex, copyList.get(randomIndex));
copyList.remove(randomIndex);
```

5- Hangi elemanın eksik olduğu bulun.

```
int returnedIndex = findMissingElement(list, copyList);
if (returnedIndex != -1) {
    System.out.println("Kopya listedeki eksik eleman: " + list.get(returnedIndex));
}

public static int findMissingElement(List<Integer> original, List<Integer> copy) {
    int missingElementIndex = -1;
    int minSize = Math.min(original.size(), copy.size());
    for (int i = 0; i < minSize; i++) {
        if (!Objects.equals(original.get(i), copy.get(i))) {
            missingElementIndex = i;
            break;
        }
    }
    // Silinen eleman döngüde bulunamadıysa orijinal listenin son indisindedir
    if (missingElementIndex == -1 && original.size() > copy.size()) {
        missingElementIndex = original.size() - 1;
    }
    return missingElementIndex;
}
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Silinecek olan sayı: [34]: 16
Kopya listedeki eksik eleman: 16

Process finished with exit code 0
```

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
Silinecek olan sayı: [98]: 6
Kopya listedeki eksik eleman: 6

Process finished with exit code 0
```