

1)JPA, Java uygulamalarındaki sınıfları ilişkisel veri tabanlarıyla eşlemek için kullanılan bir tanımlamadır. Veri tabanında tutulan bilgileri SQL sorgularıyla yönetmek yerine direkt olarak program içinde kullanılan objelerle yönetmeye olanak tanır. JPA'nın kendisi bir kütüphane değildir, bağlantıyı kuracak yapıların bunu nasıl yerine getirmesi gerektiğini gösteren bir kavramdır.

2)Türetilmiş metot isimleri iki ana parçadan oluşur ve By anahtar kelimesiyle birbirinden ayrılır. İlk kısım tanımlayıcı(introducer) olarak isimlendirilirken, ikinci kısım ölçüttür (criteria). Spring Data JPA, tanımlayıcı kısmı için find, read, query, count ve get kalıplarını destekler. Ölçüt ise oluşturulan sınıfın değişkenlerine göre belirlenir.

3)PagingAndSortingRepository, veri özelinde sayfalama ve sıralama işlemleri yapılmasına olanak sağlayan bir arayüzdür. Bütün bir verinin daha küçük parçalara ayrılarak gösterilmesi sayfalama, verinin daha düzenli bir şekilde gösterilmesi ise sıralamadır. Her iki işlem de kullanıcının veriyi daha kolay ve rahat bir şekilde anlamasına yardımcı olur.

4)getOne() metodu ilgili objenin referansını döner. Arka planda EntityManager.getReference() metodunu kullanarak lazy loading yapar. Veri bulunmaması durumunda ise EntityNotFoundException hatası yollar. findById() metodu ise objenin kendisini döner. Arka planda EntityManager.find() metodunu kullanarak eager loading yapar. Veri bulunmaması durumunda null döner.

5)@Query anotasyonu, kişiselleştirilmiş veri tabanı sorguları yazmak için kullanılır. Geliştiricinin daha karmaşık, birden çok koşullu sorgular yazarken ortaya çıkan isim karışıklığının önüne geçer.Hem SQL hem de JPQL dillerini destekler.

6)Lazy loading, bir veri yükleme yöntemidir. Veri tabanında tutulan bilgiler kullanılmak istendiğinde, sadece ihtiyaç olan verilerin kullanıcıya sunulmasıdır. Bu sayede hem performans artışı hem de kullanılan hafızanın azalması sağlanır.

7)SQL Injection, zararlı sql sorguları kullanılarak yapılan bir saldırı türüdür. Kötü niyetli kişiler bu yöntemi kullanarak veri tabanında tutulan verilere erişip, değiştirip, silebilme yetkilerine sahip olabilirler. Hibernate kütüphanesi bu saldırıyı engeller diye bir garanti de yoktur.

8)Hibernate Criteria Queries, veri tabanından verileri çekmek için kullanılan bir arayüzdür. HQL veya SQL gibi veri tabanı dilleri kullanmak yerine, nesne yönelimli programlama mantığı kullanılarak Java nesneleriyle veriler birden çok filtreleme ve mantıksal operasyondan geçirilerek kullanılabilir.

9)Erlang, fonksiyonel programlama yapılan bir yazılım dilidir. Genelde sunucular ve gömülü sistemlerde kullanılır. RabbitMQ ise Erlang kullanılarak yazılmış bir mesaj kuyruk sistemidir.

RabbitMQ, Erlang'ın sanal makinesi üzerinde çalıştığından dolayı sistemde Erlang yüklü olması gerekir.

10)JPQL, JPA'da tanımlanmış olan nesneye yönelik bir sorgulama dilidir. Veri tabanında tutulan verilerin nesne haline çevrilmiş tarzı (Entity) üzerinde işlemler yapılmasını sağlar. JPA, SQL ile JPQL arasındaki dönüşümü yaparak geliştiricilere nesneler üzerinden işlemlerini yapmasına olanak tanır.

11)İlk olarak EntityManagerFactory nesnesi oluşturulur. Oluşturulan nesneden EntityManager türünde bir nesne elde edilir. EntityManager nesnesi kayıt işlemi başlatılır ve veri tabanına kaydedilecek nesne iletilir. Kayıt bittiğinde ise işlem kapatılır.

12) 4 farklı entity mapping türü vardır:

- One-to-One: Bir yapının ilişkili olduğu yalnızca başka bir yapı olması
- One-to-Many: Bir yapının ilişkili olduğu birden fazla yapı
- Many-to-One: Birden fazla yapının ilişkili olduğu yalnızca bir yapı
- Many-to-Many: Birden fazla yapının birden fazla yapıyla ilişkisi olma durumu

13) 4 farklı entity özelliği vardır

- Persistability: Bir nesne veri tabanında saklanıp, bu nesneye her zaman erişim varsa persistent denir.
- Persistent Identity: Her bir persistent nesne benzersizdir.
- Transactionality: Entity üzerinde CRUD operasyonları yapılabilir
- Granuality: Entity ilkel veri tipi ver türlerinden oluşmamalı

14)CrudRepository, Spring Data Repository arayüzünün uzantısıdır. CRUD işlemleri yapılmasını sağlar. JpaRepository ise PagingAndSortingRepository arayüzünün bir uzantısıdır. Bu arayüz de CrudRepository arayüzünü uygular. JpaRepository hem PagingAndSortingRepository'den aldığı sayfalama ve sıralama işlemlerini, hem de kendine ait gruplayarak silme veya bütün veri tabanını ortadan kaldırmak gibi metotlara sahiptir.