

SORU 1 :

- JPA (The Java Persistence API) bir tür Java tanımlamasıdır. Java nesnesi ve onunla ilişkili veri tabanının arasındaki veriyi sürdürmek amacıyla kullanılır. JPA, nesneye yönelik alan modelleri ve onlara ilişkin veri tabanı sistemleri arasında bir tür köprü görevi görür. JPA, hangi nesnenin devam etmesi gerektiğine ve bu sürdürmenin nasıl gerçekleşeceğine yardımcı olur. JPA yalnızca bir tanımlama olduğu için herhangi bir işlemi tek başına yürütemez.

SORU 2 :

-limit – result of the query can be limited by usage of first/top keyword

findFirst10ByLastname

findFirstOrderByLastnameAsc

findTop3ByLastname

findTopOrderByAgeDesc

-property/properties expression – result will be filtered based on property of entity, multiple properties can be used with usage of And, Or keyword

findByLastnameAndFirstname

findByLastnameOrFirstname

findByFirstname

-comparison – comparison mode can be specified after specifying property used for filtering

findByFirstnames

findByFirstnameEquals

findByStartDateBetween

findByAgeLessThan, findByAgeLessThanEqual

findByAgeGreaterThan, findByAgeGreaterThanEqual

findByStartDateBefore, findByStartDateAfter

findByAgeIsNull, findByAgeIsNotNull

findByFirstnameLike, findByFirstnameNotLike

findByFirstnameStartingWith, findByFirstnameEndingWith

findByFirstnameContaining

findByLastnameNot

findByAgeIn(Collection ages), findByAgeNotIn(Collection ages)

findByActiveTrue, findByActiveFalse

findByFirstnameIgnoreCase

-ordering operator – optionally you can specify ordering operator at the end of method name

findByLastnameOrderByFirstnameAsc

findByLastnameOrderByFirstnameDesc

SORU 3 :

Veri havuzunu sayfalandırmak ve sıralamak için PageRequest nesnesine ihtiyaç vardır. PageRequest nesnesi, istenilen sayfa numarası ve boyutuna geçiş yapılarak yaratılabilir.

-Pageable firstPageWithTwoElements = PageRequest.of(0, 2);

-Pageable secondPageWithFiveElements = PageRequest.of(1, 5);

Aynı zamanda Spring Data Web Project kullanılarak da denetleyici 'pageable' örneğini elde edilebilir.

-Page<Product> allProducts = productRepository.findAll(firstPageWithTwoElements);

List<Product> allTenDollarProducts =

productRepository.findAllByPrice(10, secondPageWithFiveElements)

Sıralamak için;

-Page<Product> allProductsSortedByName = productRepository.findAll(Sort.by("name"))

Hem sayfalandırmak hem sıralamak için;

-Pageable sortedByName =

PageRequest.of(0, 3, Sort.by("name"));

-Pageable sortedByPriceDesc =

PageRequest.of(0, 3, Sort.by("price").descending());

-Pageable sortedByPriceDescNameAsc =

PageRequest.of(0, 5, Sort.by("price").descending().and(Sort.by("name")))

SORU 4:

findById	findOne
findById() methodu CrudRepository arayüzünde kullanıma uygundur.	findOne methodu JpaRepository arayüzünde kullanıma uygundur.
-findById() methodu eğer kayıt veri tabanında bulunmuyorsa null'a dönüşür.	- findOne methodu kayıtların veri tabanında bulunmama durumunda EntityNotFoundException yanıtını verir
İçten findById() methodu EntityManagerfind methodunu kullanır.	-İçten findOne methodu EntityManagergetReference() methodunu kullanır
findById()'u çağırarak istekli olarak getirilmiş entity'e döner.	-findOne()'ı çağırarak tembelce getirilmiş entity'e dönüşür.
-findById() methodları asıl nesnelere dönüşür ve entity dosyaları veri tabanından değer içerir.	-findOne entity'nin referansına dönüşür.Bütün dosyalar default değerleri içerebilir.

SORU 5 :

@query veritabanında birçok işlem gerçekleştirebilir. En bilinen fonksiyonu ise çizelgelerden spesifik verileri geri almaktır.

SORU 6 :

Lazy loading uyku durumundaki bütün entityler için kullanılan bir tür getirme tekniğidir.Veli sınıfı nesnesi yüklerken çocuk sınıfı nesnesi kullanıp kullanılmayacağına karar verir.

SORU 7 :

-SQL enjeksiyonu, saldırganların veri tabanına uygulanan sorgulara müdahale etmesine izin veren bir tür web güvenlik hassasiyetidir. Genellikle saldırganların normalde geri alamadıkları verileri görmesine izin verir.

SORU 8 :

Uyku modundaki API ölçütleri dinamik olarak ölçüt sorgusu oluşturmaya yarar.Ölçüt, HQL ve ana SQL sorgularından farklı bir veri getirme tekniğidir.API ölçütünün en önemli faydası zor yazılmış SQL ifadelerine gerek kalmadan veriyi manipüle edebilmesidir.

SORU 9 :

-Erlang bir tür kodlama dilidir.Aynı anda kullanım, dağıtım ve hata toleransına destek için yaratılmıştır.

SORU 10 :

JPQL (Java Persistence Query Language), süregelen oluşumlarda veritabanı işlemleri gerçekleştirmek için kullanılan, nesneye yönelik bir sorgu dilidir.

SORU 11 :

- 1- Yönetici fabrika/atölye entity nesnesi yaratmak.
- 2- Atölyeden entity yöneticisi edinmek.
- 3- Oluşum yöneticisini başlangıç durumuna getirmek.
- 4- İlişkili veri tabanında veri sürmek.
- 5- İşlemleri kapatmak.
- 6- Atölye kaynaklarını yayımlamak.

SORU 12 :

- 1)Opis\ORM\IMappableEntity
- 2)the construct-time of the entity manager
- 3)registerMappingCallback

1- use Opis\ORM\{

Entity, IMapper, IMappableEntity

};

class User extends Entity implements IMappableEntity

{

public static function mapEntity(IMapper \$mapper)

{

// Map entity here

}

}

2-use Opis\Database\Connection;

use Opis\ORM\{

EntityManager, IMapper

};

use MyBlog\{

User, Article

};

// Define a database connection

```
$connection = new Connection("dsn:mysql;dbname=test", "root", "secret");
```

```
// callbacks
```

```
$mappers = [
```

```
    User::class => function(EntityMapper $mapper) {
```

```
        // map entity here
```

```
    },
```

```
    Article::class => function(EntityMapper $mapper) {
```

```
        // map entity here
```

```
    }
```

```
];
```

```
// Create an entity manager
```

```
$orm = new EntityManager($connection, $mappers);
```

```
3-$orm->registerMappingCallback(User::class, function(EntityMapper $mapper){
```

```
    // map entity here
```

```
});
```

SORU 13:

Veri değeri, bir ya da daha fazla özelliğe sahip entitylerle bağlantılıdır. Her bir özelliğin ismi ve bir veya birden daha fazla değeri vardır.

SORU 14:

CrudRepository genellikle CRUD işlemlerini temin eder. JpaRepository ise JPA bağlantılı methodları temin eder.