

3. الـمجال نموذج تعريفي

يتم حيث التـقاليدي سـتخدم الـمجال الـبيانات قاعدة في جدول لـتمثيل بـسيطة لكي ان فئـة إنـشاء الـأعمدة. إلى الـحقول الـمجال الـصغري، بالـخط افـتراضي بـشكل الـجدول اسم إلى الـفئـة اسم الـمجال.

```
import org.springframework.data.annotation.Id;

public class Person {

    @Id
    private Long id;
    private String firstName;
    private String lastName;

    // Constructor (Spring Data JDBC)
    public Person() {}

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    // Getters and setters
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getFirstName() { return firstName; }
    public void setFirstName(String firstName) { this.firstName = firstName; }
    public String getLastName() { return lastName; }
    public void setLastName(String lastName) { this.lastName = lastName; }
}
```

الـأساسي. الـمفتاح يـحدد @Id
الـمجال الـبيانات قاعدة في جدول لـتمثيل بـسيطة لكي ان فئـة إنـشاء الـأعمدة.
ذلك. تـجاوز تـم إذا إلـ person الـجدول تـسميـة سـي تـم

4. مـسـتودع إنـشاء

الـأساسي الـعمليات لـتـسـجيل CrudRepository تـم تـد واجهـة تعريـف

```
import org.springframework.data.repository.CrudRepository;

public interface PersonRepository extends CrudRepository<Person, Long> {
}
```

التشغيل. وقت في التنفيذي ولد `PersonRepository` `Person` `Person` تنفيذه إلى تحتاج ل شيء! كل هذا

5. المستودع استخدام

واستخدمة: `PersonService` أو خدمة في المستودع بإدخال م

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class PersonService {
    private final PersonRepository repository;

    @Autowired
    public PersonService(PersonRepository repository) {
        this.repository = repository;
    }

    public void savePerson() {
        Person person = new Person("John", "Doe");
        repository.save(person);
        System.out.println("Saved person with ID: " + person.getId());
    }

    public void listPeople() {
        Iterable<Person> people = repository.findAll();
        people.forEach(p -> System.out.println(p.getFirstName() + " " + p.getLastName()));
    }
}
```

6. تطبيقي ك تشغيل

بتشغيله: قم و `@SpringBootApplication` مع رئيسية فئة بإنشاء قم، `PersonRepository` `Person` `Person` تستخدم لنت إذا

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {
    public static void main(String[] args) {
```

```

        SpringApplication.run(Application.class, args);
    }
}

```

تلقائيًا. `PersonRepository` و `Person` المصدر بتكويّن سيقوم `PersonRepository`

7. مخصصة استعلامات اختياري:

`@Query` مع مستودعك إلى طرق إضافة يمكنك مخصصة، استعلامات إلى تحتاج كنت إذا

```

import org.springframework.data.jdbc.repository.query.Query;
import org.springframework.data.repository.CrudRepository;

public interface PersonRepository extends CrudRepository<Person, Long> {
    @Query("SELECT * FROM person WHERE first_name = :name")
    List<Person> findByFirstName(String name);
}

```

تذكره يجب التي الرئسيّة النقاط

- العددي إلى واحد مثل العلاقات المتأخر. التحميل يدعم `Person`، `Person` عكس على متأخر: تحميل يوجد `Person` التجمعات. جذور مع صريحة نموذجة إلى تحتاج
- مباشرة. استرجاعه أو حفظه يتم الجذر فقط و تجمع، من جزء لكيان لكل التجمع: جذر
- عبر المثال، سبيل على الهيكل توفير على تلقائيًا. الجداول يخلق `Person` `Person` الهيكل: إنشاء `schema.sql` في `src/main/resources`:

```

CREATE TABLE person (
    id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255)
);

```

العمل سير على مثال

1. تطبيقك. ابدأ
2. `repository.save()` باسم `Person` كائن احفظ.
3. `repository.findById()` أو `findAll()` باسم `Person` استرجعه.

وسأقوم هدفك، هو ما لي فقل معني، شيء أو المخصصة، الم `Person` أو العلاقات، في الغوص تريد كنت إذا الأساس! هو هذا الآن؟ حتى هذا يعمل كيف التفسير. بتخصيص