

## @Entity

- ▶ Nos permite indicar que una clase Java será una entidad.
- ▶ Esta podrá ser trasladada a una BDR como una tabla.
- ▶ Nos permite trabajar con instancias de la clase como si fueran filas de una tabla.

# Definición de una entidad

```
@Entity
@Table(name="STUDENT")
public class Student {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    @Column(name="studentId")
    private long id;
    @Column
    private String first_name;
    @Column
    private String last_name;

    //Getters, setters, ...

}
```

La clase será tratada como una entidad

Será mapeada como una tabla en la BD, llamada STUDENT

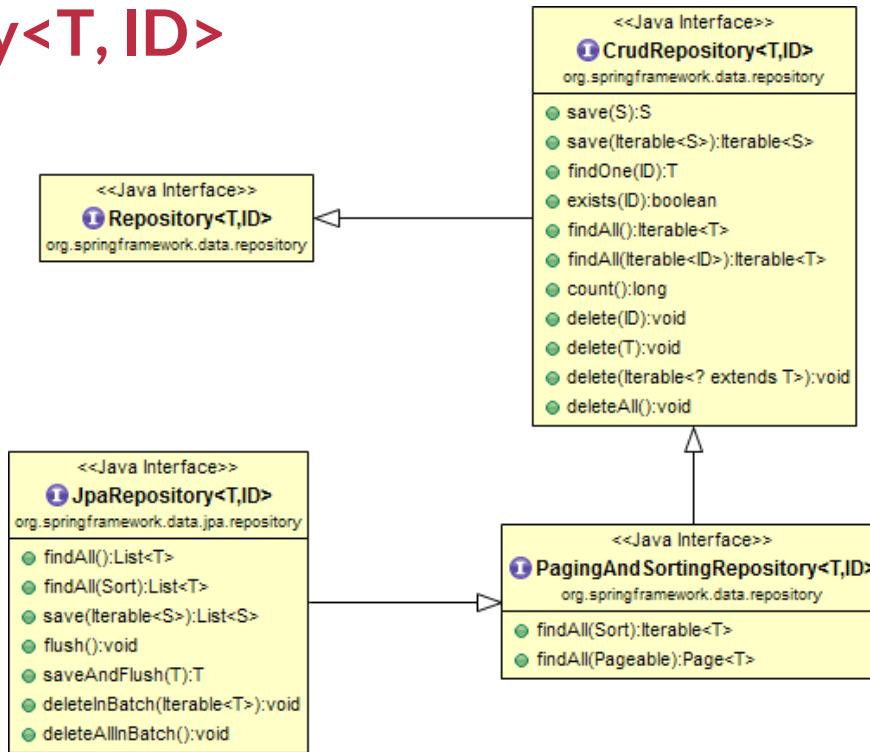
Representa la PK de la tabla. Se autogenerará

Modificamos el nombre por defecto de la columna

Otras columnas de la tabla en la base de datos

# JpaRepository<T, ID>

- Spring Data proporciona varios tipos de repositorios.
- JpaRepository es el más completo que podemos usar con JPA



## @Repository

- ▶ Nos permite indicar que una interfaz será tratada como un repositorio
- ▶ Spring lo podrá localizar mediante la anotación `<jpa:repositories... >`

# Definición de un repositorio

```
package com.openwebinars.springmvc.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.openwebinars.springmvc.model.Student;

@Repository
public interface StudentRepository extends JpaRepository<Student, Long>{

}
```

Anotación que define el repositorio

Tipos parametrizados sobre los que definimos nuestro repositorio