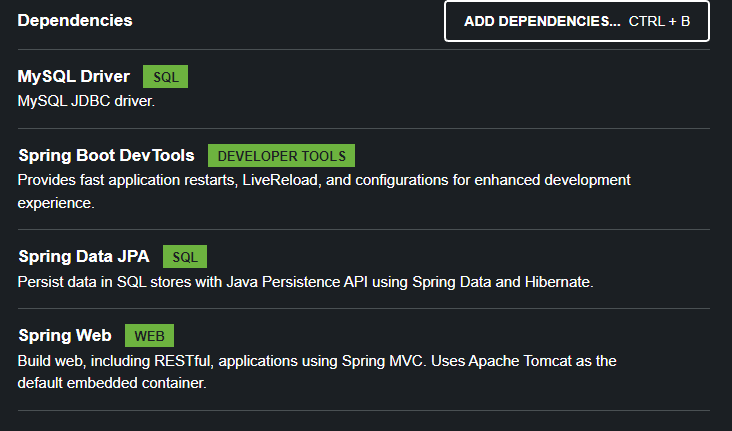
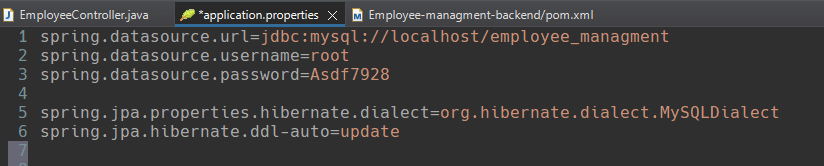
1. Creamos el proyecto desde la web de Sprinboot y agregamos las dependencias que aparecen en pantalla
2. Desde MYSQL vamos a crear la base de datos y comenzar a utilizarla con los siguientes comandos

Create database “nombre de la db”

Use “nombre de la db”

(En este caso sería “create database employee managment”)

1. En Eclipse vamos a src/main/resources > application.properties y vamos a configurar la DB



\*1 – Es la dirección de url a la base de datos. En este caso es localhost y la siguiente sentencia es el nombre de la db

\*2 – Creería que es el nombre del user de MYSQL, lo estoy probando

\*3 – Mismo

\*5 – Hace referencia al dialecto utilizado  (Hay que revisar que sea compatible con la version deMySQL)

\*6 – Hace que cuando se abra la app lo primero que haga sea el update

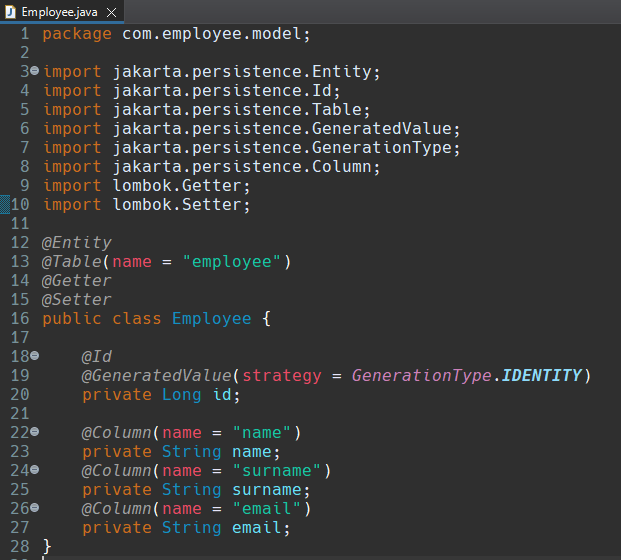
4 – En la carpeta src/main/java creamos el paquete MODEL para crear las clases a utilizar. En este caso Employee   
A la clase le agregamos las anotaciones @Entity y @Table (name = “employee”). Importamos ambas anotaciones (en name = employee es para marcar a que tabla de la DB va a hacer referencia)   
Le agregamos ya dentro de la clase:

@Id   
@GeneratedValue(stratefy = GenerationType.IDENTITY)

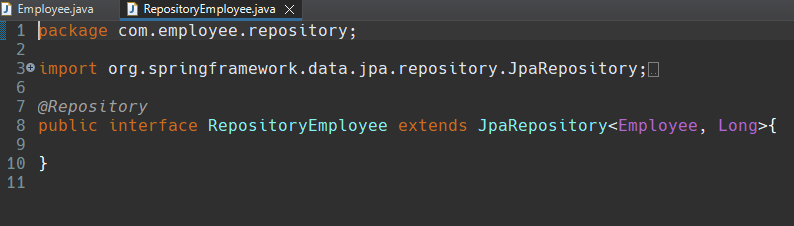
El ID es para marcar que va a ser la primary key y el generated... es para hacer que se auto incremente

Luego creamos los atributos privados e importamos la dependencia lombok para poder utilizar las anotaciones de getter y setter (los getters y setters tienen que estar por encima de la public class Employee(

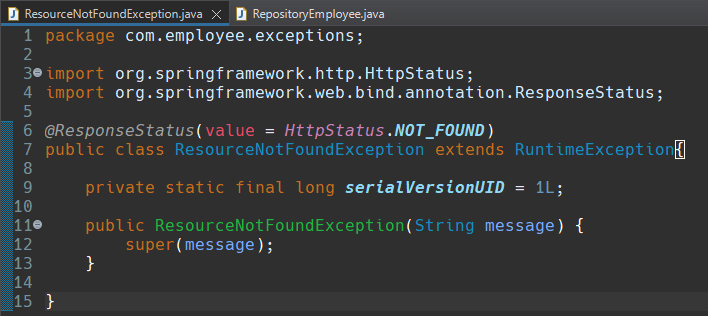
<**dependency**>   
 <**groupId**>org.projectlombok</**groupId**>   
 <**artifactId**>lombok</**artifactId**>    
 <**scope**>provided</**scope**>    
</**dependency**>



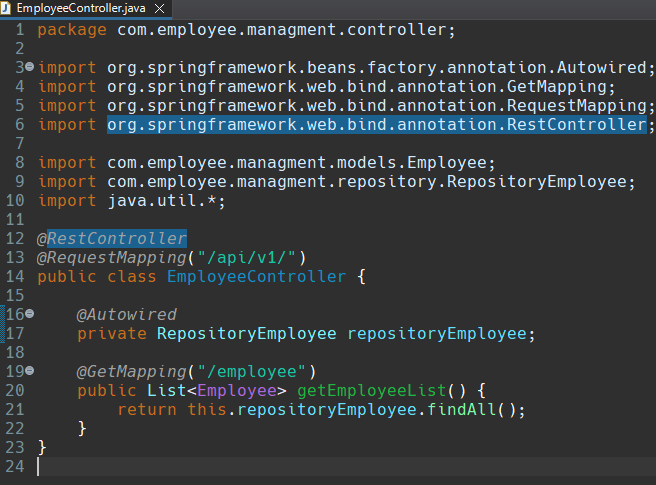
5 – En la carpeta src/main/java creamos el paquete com.employee.repository y creamos la interfaz RepositoryEmployee   
A la interfaz, le agregamos la anotacion de @Repository   
Luego le agregamos “extends JpaRepository<Employee, Long>”   
Employee va a ser la clase que trabaja y Long es el ID o key   
Importamos la herencia de Jpa



6 – Ahora vamos a crear el paquete para la excepciones “com.employee.exceptions” (tambien  dentro de main/java)   
Creamos la clase ResourceNotFoundException   
Le agregamos la anotación @ResponseStatus(value HttpStatus.NOT\_FOUND) y le agregamos la extension de RuntimeException   
Creamos un atributo privado static final long “serialVersionUID = 1L;”   
Luego creamos el constructor que recibe un String “message” desde la clase super



7 – Creamos el paquete controller y adentro la clase EmployeeController.   
Le agregamos la anotación @RestController y @RequestMapping(“/api/v1/”)   
Creamos un atributo del tipo RepositoryEmployee y le agregamos la anotación @Autowired (inyección) para que sepa que esa información va a ser inyectada  
Luego creamos el metodo getEmployeeList() que retorna repositoryEmployee.findAll(); esto nos devolver la lista complete de los empleados. Se le agrega la anotación @GetMapping(“/employee”) para que en la url nos muestre el retorno (en este caso es la lista de empleados)



* + EN ESTA INSTANCIA YA PODEMOS PROBAR EL FUNCIONAMIENTO DE LA APP  
    VAMOS A MYSQL Y AGREGAMOS MANUALMENTE DOS EMPLEADOS Y DESDE LA URL <http://localhost:8080/api/v1/employee> VAMOS A VER LOS RESULTADOS

8 -