

Ejercicios Curso JPA

Formación Sopra Steria
Formación JPA

Bloque JPA

Versión 1.0 del martes, 14 de diciembre de 2018

Historial

Versión	Fecha	Origen de la actualización	Redactado por	Validado por
1.0	14/12/2018		Alba Bermejo Solís Adrián Colmena Mateos Emilio Guillem Simón Nicole Tarela Duque Adrián Verdú Correcher Alejandro Mus Mejías	



Índice

1.	Ejercicios Capítulo 2: Introducción a entidades y persistencia	4
2.	Ejercicios Capítulo 3: ORM	5
3.	Ejercicios Capítulo 4	7



1. Ejercicios Capítulo 2: Introducción a entidades y persistencia

1. **CREAR UNA ENTIDAD.** Se pide crear una nueva Entidad "Planta". Esta entidad tendrá como atributos un id, que actuará como primary key en la base de datos, un nombre y un tipo.
2. **CREAR TABLA EN LA BASE DE DATOS HR.** Se deberá crear una tabla con el mismo nombre y atributos que la entidad Planta en la base de datos hr.
3. **CREAR UN SERVICIO.** Para trabajar con la entidad Planta, se debe crear una clase "ServicioPlanta" que opere con ella. En primer lugar, se pide hacer un método para crear y persistir registros de tipo Planta.
4. **MÉTODO MAIN.** Se creará una clase llamada "ProgramaPlanta" que contará con un método main. Este método se ocupará de llamar al ServicioPlanta y sus métodos. En primer lugar, debe crear un registro de tipo Planta en la base de datos y pintarlo en consola.
5. **BUSCAR.** Se deberán crear dos métodos nuevos en el ServicioPlanta: buscarPlanta y buscarTodasLasPlantas. El método buscarPlanta pedirá el id de una Planta y la buscará, devolviendo el objeto Planta encontrado. El método buscarTodasLasPlantas listará todas las Plantas en la base de datos. Una vez creados se llamarán desde el método main de la clase ProgramaPlanta y se pintarán los resultados en consola.
6. **MODIFICAR Y BORRAR.** Se deberán crear dos métodos nuevos en el ServicioPlanta: cambiarTipo y borrarPlanta. El método cambiarTipo seleccionará el registro Planta seleccionado por su id y modificará el atributo Tipo. El método borrar buscará el registro Planta especificado por el id y lo borrará de la base de datos. Ambos métodos deben ser llamados por el método main de la clase ProgramaPlanta y pintar un mensaje en pantalla que especifique si se ha realizado la acción de manera correcta.



2. Ejercicios Capítulo 3: ORM

1. **CREAR ENTIDADES RELACIONADAS.** A partir de la base de datos *hr* de Oracle y haciendo uso de sus tablas, se pide crear las entidades *Employee* y *Department* con relación de *OneToMany* desde *Department* y *ManyToOne* desde *Employee*.

EMPLOYEE

- 1.1. Los nombres de los atributos serán diferentes a los de los campos de la base de datos.
 - 1.2. Además de los imprescindibles, se tendrán en *Employee* los atributos referidos a los campos: `FIRST_NAME`, `PHONE_NUMBER`, `SALARY` y `DEPARTMENT_ID`.
 - 1.3. El atributo que hace referencia a `HIRE_DATE` será de tipo `Date` del paquete `java.util.Date`. Para ello utilizar la anotación conveniente.
2. El atributo que hace referencia a `EMAIL` será *lazy loaded*. Utilizar la anotación conveniente.
- 2.1.

DEPARTMENT

- 2.2. Los nombres de los campos de Java (atributos) serán iguales a los de los campos de la base de datos. Es decir, no se utilizará la anotación `@Column`.
 - 2.3. Tendrá solo los atributos imprescindibles.
3. Se utilizará el acceso mixto de la siguiente manera: Si la longitud del nombre del departamento es menos de 10 caracteres, se le añadirá al principio "dep. ", y si no, se quedará el nombre original. Esto se hará en el método *getter* de `department_name`. Utilizar las anotaciones convenientes.
4. **CREAR SERVICIOS Y UN MÉTODO MAIN.** Para trabajar con las entidades, se deben crear las clases *ServiceEmployee* y *ServiceDepartment* que operen con cada una de ellas. Hacer los métodos que se consideren necesarios para comprobar el correcto funcionamiento de las entidades. Como mínimo, un método que cree un departamento, otro que cree un empleado que pertenezca a ese departamento y otro que busque un empleado con un id determinado.
5. **UTILIZAR LA GENERACIÓN DE IDENTIFICADORES MEDIANTE SECUENCIAS.** Utilizar las secuencias dadas por la base de datos (`EMPLOYEES_SEQ` y `DEPARTMENTS_SEQ`) para la generación de identificadores de ambas entidades asociados a `EMPLOYEE_ID` asociados a `EMPLOYEE_ID` y `DEPARTMENT_ID`. Utilizar las anotaciones convenientes. Volver a comprobar la creación de empleados y/o departamentos y verificar que en la base de datos los identificadores se incrementan de forma correcta.



6. **CREAR UN OBJETO EMBEBIDO.** Añadir el atributo referido a `MANAGER_ID` de la base de datos a ambas entidades como objeto embebido. Para ello, crear una nueva clase *Manager*. Utilizar las anotaciones necesarias. Cambiar el método *main* para crear otro empleado y otro departamento pasándole un manager y comprobar su correcto funcionamiento.



3. Ejercicios Capítulo 4

En **TODOS** estos ejercicios se va a trabajar con el esquema hr de la base de datos de Oracle. En cuestión, se utilizarán las tablas "COUNTRIES" y "REGIONS" para practicar el uso de consultas JP QL.

1. **MAPEO DE ENTIDADES.** Se pide crear un nuevo proyecto JPA, en el que se crearán dos nuevas entidades llamadas "Countries" y "Regions", que deben ser relacionadas adecuadamente entre ellas y referenciadas a las tablas homónimas del esquema hr de la base de datos.
2. **LISTADO DE TODOS LOS REGISTROS.** Se deben crear dos nuevas clases "ServicioCountries" y "ServicioRegions" en las que se crearán los métodos asociados a todas las consultas. Para empezar, se crearán dos métodos para listar todas las instancias de las entidades "Countries" y "Regions". Estos métodos deben utilizar la interfaz *TypedQuery* para realizar la consulta que devuelva el resultado deseado. Se creará además una clase Programa con un método *main* que llamará a estos métodos y mostrará por pantalla los resultados de las consultas.
3. **CLAÚSULAS JOIN Y WHERE.** Se debe generar una consulta con nombre que devuelva por pantalla todos los países que pertenecen a la región Europa, así como los datos de la región.
4. **EXPRESIONES BETWEEN, LIKE, IN Y SUBCONSULTAS.** Se debe generar una consulta con nombre que devuelva los países cuyos nombres terminen con las letras "ia" y que pertenezcan a regiones cuyo ID se encuentre entre 3 y 4. Para ello se deberán utilizar las expresiones BETWEEN, LIKE, IN y realizar una subconsulta.
5. **CONSULTAS AGREGADAS.** Se debe realizar una consulta con nombre que muestre el nombre de las regiones cuyos id se encuentren comprendidos entre 1 y 3. Además se deberá mostrar también el número de países pertenecientes a esa región. Se deberán agrupar los resultados por el nombre de la región, ordenarlos también por el nombre de la región y, por último, sólo se mostrarán aquellas regiones que tengan un número de países mayor que 5.
6. **UPDATE.** Para este ejercicio se deberá crear una nueva instancia en la tabla "REGIONS", de la base de datos, llamada "Sistema Solar", mediante código SQL. Se crearán también instancias en la tabla "COUNTRIES" que aludan a los planetas pertenecientes a la región "Sistema Solar". Se creará una consulta con nombre en el programa que modifique el nombre de todos los planetas, añadiendo al nombre la cadena "_Planeta". Una vez ejecutada la consulta, se comprobará que se hayan realizado los cambios en la base de datos.



7. **DELETE.** En este ejercicio se trabajará de nuevo con la región "Sistema Solar" y los planetas. Se creará una consulta con nombre que elimine todos los planetas y otra consulta con nombre que elimine la región "Sistema Solar". Una vez ejecutadas las consultas, se comprobará que se hayan realizado los cambios en la base de datos.

