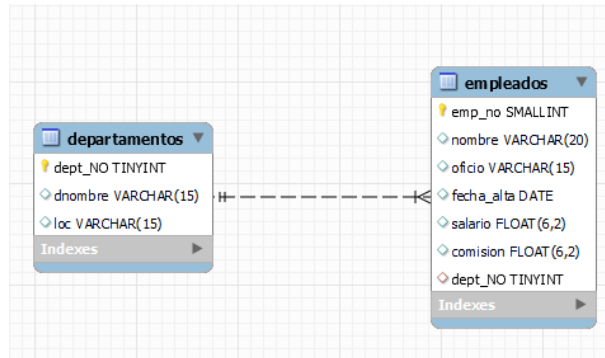


Práctica3: Consultas HQL.

Sobre la base de datos `ejemploh` utilizada en la práctica 1



Utiliza el proyecto Maven creado en la práctica 1 (**p1_hibernate**).

- El `group id` del proyecto deberá ser `org.iesabastos.dam.datos.xxx`, donde `xxx` son tus iniciales (nombre, apellido1, apellido2)
- El `artifact id` del proyecto será `p1_hibernate`
- Añade las dependencias necesarias para trabajar con Hibernate y MySQL
- Crea los POJOs para las tablas `empleados` (`Empleado.java`) y `departamentos` (`Departamento.java`)
- Mapea la relación 1 – n entre `departamentos` y `empleados`
- Crea la clase `HibernateUtil.java` como se ha visto en clase
- Crea el fichero de configuración (`hibernate.cfg.xml`)

A continuación, crea una nueva clase llamada **_08_ConsultasHQL** que realice sentencias HQL para obtener la siguiente información de la base de datos:

- 8.1. La información de cada departamento
 - 8.2. El nombre de los departamentos
 - 8.3. El nombre y localización de los departamentos.
 - 8.4. Los empleados dados de alta en 2016.
 - 8.5. El nombre de los empleados del departamento de Informática.
 - 8.6. ¿Cuántos departamentos hay en el instituto?
 - 8.7. Cuántos empleados tenemos en cada departamento (de aquellos departamentos que tienen empleados)
 - 8.8. Los departamentos que no tienen empleados.
 - 8.9. Los departamentos que no tienen ningún empleado que se llame Cristina.
- Nota. Si tienes pocas filas añade previamente más filas a tu base de datos.

9. Crea una nueva clase, llámala **_09_QueryDep** para recuperar una lista de los departamentos utilizando el método `list()` y después el método `iterate()`.
10. Crea una nueva clase **_10_QueryParam** para recuperar los empleados cuyo sueldo sea igual superior a 800 euros utilizando parámetros.
11. Crea una nueva clase **_11_MasAntiguo** para obtener el empleado con más antigüedad y saca una lista de todos sus compañeros de departamento. La lista de los compañeros se obtendrá mediante una consulta con parámetros.
12. Crea una nueva clase **_12_QueryClasesNoAsociadas** para obtener el código y nombre de los departamentos junto con los empleados ordenados por los nombres de los empleados con `iterate()`
 - 12.1. Rectifica la clase utilizando ahora `iterate()` en vez de `list()`.
13. Crea una nueva clase **_13_QueryFuncionesGrupo** para obtener obtener el salario medio de los empleados.
 - 13.1. Sobre la consulta anterior añade lo necesario para obtener ahora el salario medio y el nº total de empleados.