

Hibernate

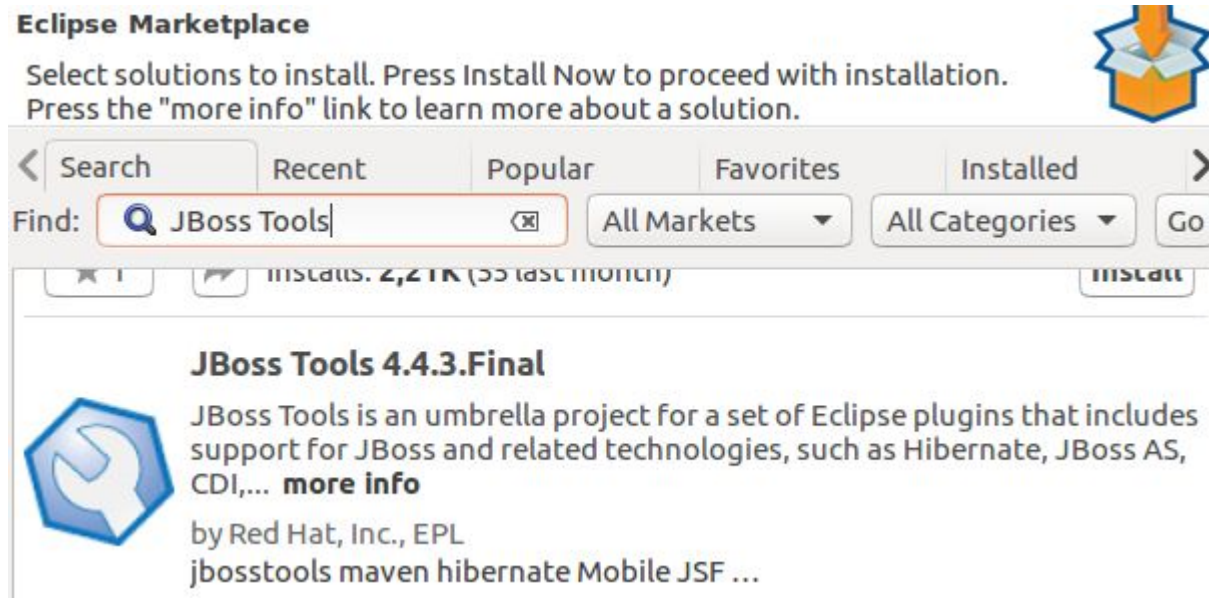
Instalación y configuración en Eclipse

Instalación del plugin

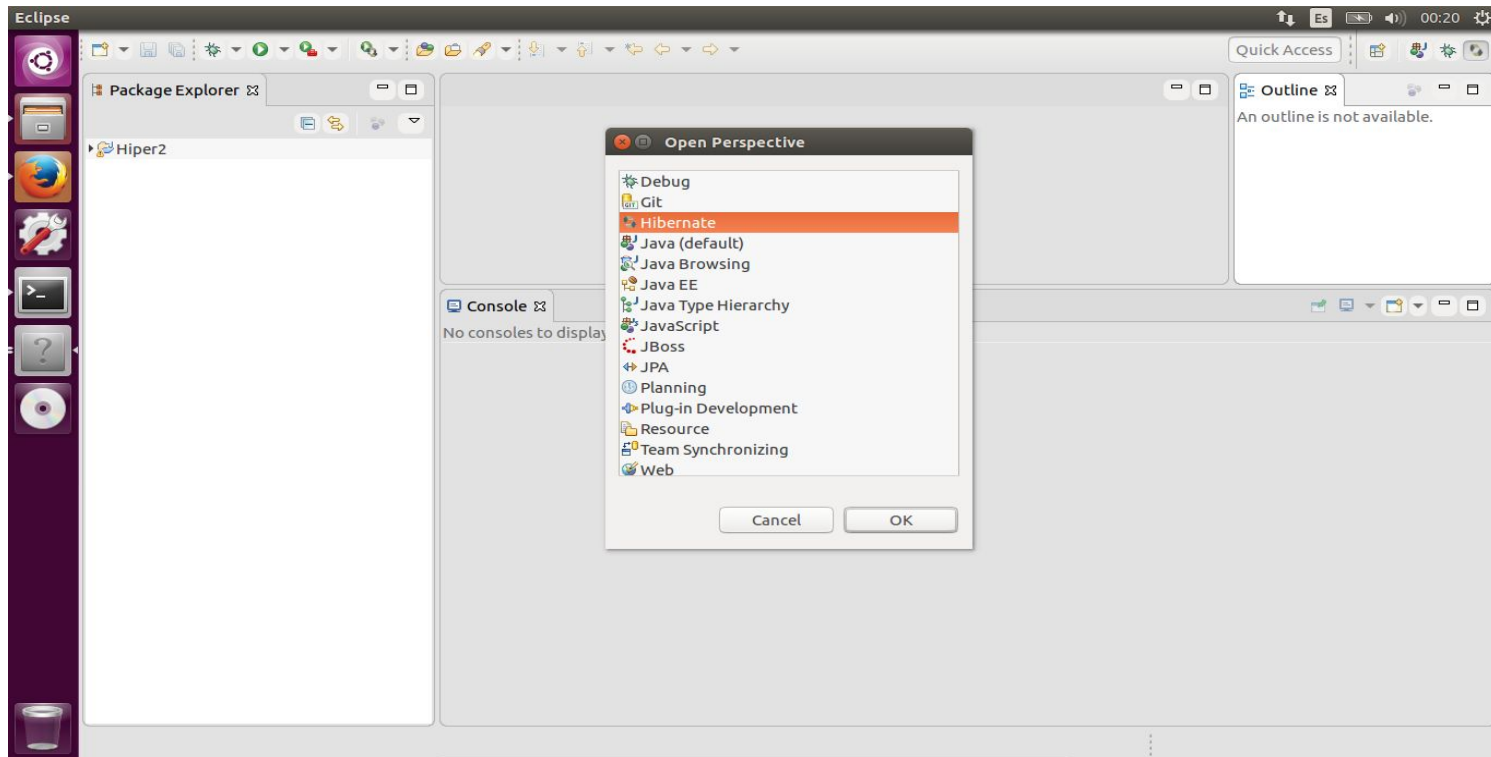
- En primer lugar debemos saber qué versión de Eclipse disponemos.



- Vamos a instalar el plugin para Eclipse de “JBoss Tools”
- Podemos acceder al [web](#) de JBoss para averiguar cómo debemos instalar el plugin. Debe ser el plugin acorde con nuestra versión de Eclipse.
- Después, en Eclipse, accedemos al menú **Help** → **Eclipse Marketplace**.
Buscamos el plugin **JBoss Tools** y lo instalamos (Requerirá reiniciar Eclipse)



- Comprobamos que se ha instalado correctamente el plugin. Para ello **Window** → **perspective** → **other**, entre las opciones que aparecen debe estar la perspectiva **Hibernate**.



Configuración del driver MySql

- Vamos a trabajar con la base de datos MySql. Necesitaremos tener el conector accesible y bien configurado.
- En primer lugar, nos bajamos el conector JDBC desde [aquí](#). Se trata del conector de Tipo4 con el que ya hemos trabajado en clases anteriores.
- Ahora, en Eclipse, hacemos **Window** → **preferences**. Y en el menú de la izquierda escogemos **DataManagment** → **Connectivity** → **Driver definitions**. A continuación debemos clicar en “Add” para añadir el nuevo conector. Nos aparecerá un cuadro de diálogo rotulado “**New Driver Definition**” (Observa siguiente figura)



type filter text

Driver Definitions

New Driver Definition

Specify a Driver Template and Definition Name

✖ A driver already exists with that name. Please provide a unique driver name.

Name/Type JAR List Properties

Vendor Filter: All

Available driver templates:

Name	System Vendor	System Version
Database		
Generic JDBC Driver		

Driver name:

MySQL Driver

Driver type:

Generic JDBC Driver



Cancel

OK

Add...

Edit...

Remove

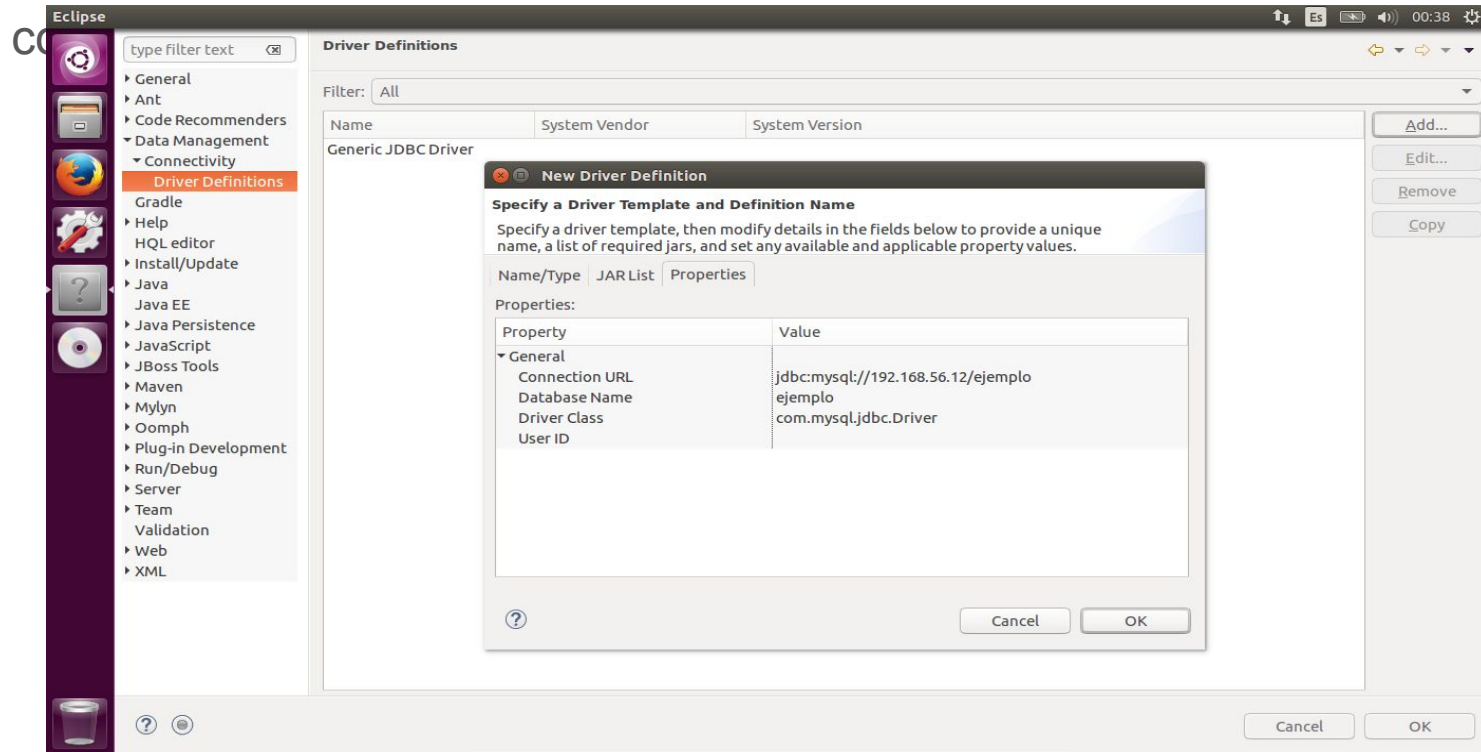
Copy



Cancel

OK

- La ventana que nos aparece dispone de tres pestañas. La primera (**Name/Type**) nos pide un nombre para el driver, debemos especificar quién es el Vendor y el tipo de conector. En la segunda (**JAR list**) debemos indicar la ruta al paquete .jar del conector. Y en la tercera (**Properties**), debemos



- Para configurarlo en la pestaña **properties** los valores dependen de cómo tengamos nosotros configurada la base de datos. En principio utilizaremos la base de datos “ejemplo” que venimos usando desde hace ya varias clases.

- Campo URL → Indicamos el localizador del recurso. Tendrá una forma similar a:

```
jdbc:mysql://<IP servidor MySql>/<Nombre bbdd>
```

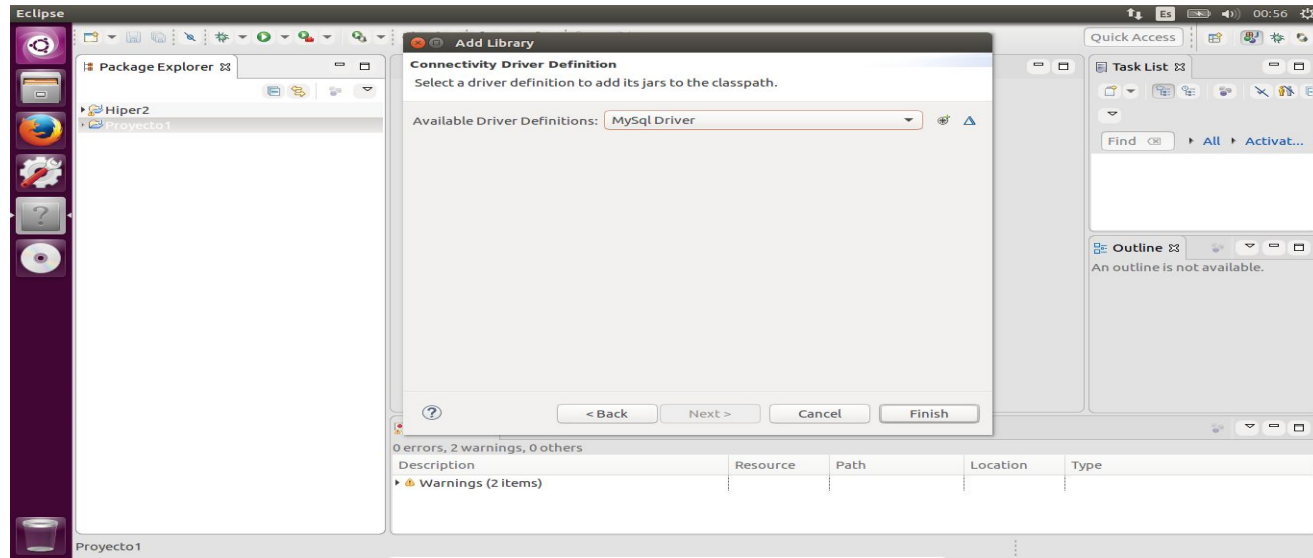
- Campo Database Name → Indicamos el nombre de la bbdd. “ejemplo” en mi caso.
- Campo Driver Class → Indicamos la clase del paquete que utilizaremos como driver

```
com.mysql.jdbc.Driver
```

- Campo User ID → No indicamos nada.

Primer proyecto

- Creamos un nuevo proyecto Java en Eclipse. Lo llamaremos Proyecto1
- Seleccionamos el nuevo proyecto y clicando el botón derecho **Build Path** → **Add Libraries**. Se visualiza una ventana, elegimos **Connectivity Driver Definition** y, tras el *Next* correspondiente elegimos el driver recién definido.



Configuración Hibernate

- Una vez tenemos la librería MySQL en nuestro proyecto hemos de crear el fichero de configuración de Hibernate llamado **hibernate.cfg.xml**
- Seleccionamos nuestro proyecto, botón derecho y **New** → **Other** → **Hibernate** → **Hibernate Configuration File (cfg.xml)**. Este fichero es un XML que contiene todo lo necesario para realizar la conexión a la base de datos.
- Pulsamos *Next*. A continuación, nos pide dónde crear el fichero. Lo haremos en la carpeta **src** del proyecto. Otro *Next* y pasamos a configurar la conexión.

- Los campos a rellenar son:
 - **Session factory name** → nombre de la conexión. Pondremos “Miconexion”.
 - **Database dialect** → elegimos tipo de comunicación JDBC. En concreto, opción “MySQL”
 - **Driver Class** → clase JDBC para la conexión → **com.mysql.jdbc.Driver**
 - **Conection URL** → ruta de conexión a nuestra base de datos. Es la misma que hemos indicado cuando definíamos el driver
 - **Username** → Usuario que se conectará a MySql. OJO: debe tener permisos (Debe estar comentado, por cierto, el *bind address* famoso...)
 - **Password** → Contraseña correspondiente.
- Puedes observar la siguiente diapositiva para disponer de una referencia
- Para terminar pulsamos *Finish*. Aparece el editor de configuración de Hibernate. Desde la pestaña **Source** se puede editar el fichero XML generado.



Package Explorer

Hiper2
Proyecto1
src
JRE System Library [JavaSE-1.8]
MySQL Driver

Proyecto1

Hibernate Configuration File (cfg.xml)

This wizard creates a new configuration file to use with Hibernate.

Container: /Proyecto1/src
File name: hibernate.cfg.xml
Hibernate version: 5.2
Session factory name: Miconexion
Get values from Connection
Database dialect: MySQL
Driver class: com.mysql.jdbc.Driver
Connection URL: jdbc:mysql://<hostname>/<database>
Default Schema:
Default Catalog:
Username: user
Password: austria

☐ Create a console configuration



< Back

Next >

Cancel

Finish

Quick Access

Task List

Find

All Activat...

Outline

An outline is not available.

ation

Type

- Una vez creado el **hibernate.cfg.xml**, hemos de crear el fichero **XML Hibernate Console Configuration**.
- Seleccionamos nuestro proyecto → botón derecho → **New** → **Other** → **Hibernate** → **Hibernate Console Configuration**.
- Correspondiente *Next* y en el campo Name de la nueva ventana escribimos un nombre para nuestra configuración de consola Hibernate. Por ejemplo “Proyecto1ConsoleConfiguration” (observa siguiente diapositiva)
- Clicamos *Finish*

Package Explorer

Hiper2

Proyecto1

src

JRE System Library [JavaSE-1.8]

MySQL Driver

Create Hibernate Console Configuration

This wizard allows you to create a configuration for Hibernate Console.

Name: Proyecto1ConsoleConfiguration

Main Options Classpath Mappings Common

Type:

☒ Core ☐ Annotations (jdk 1.5+) ☐ JPA (jdk 1.5+)

Hibernate Version: 5.2

Project:

Proyecto1

Browse...

Database connection:

[Hibernate configured connection]

New...

Edit...

Property file:

Setup...

Configuration file:

/Proyecto1/src/hibernate.cfg.xml

Setup...

Persistence unit:

Browse...



< Back

Next >

Cancel

Finish

Quick Access

Task List

Find

All Activat...

Outline

An outline is not available.

ation

Type

- Por último, para acabar de configurar Hibernate, hemos de crear el fichero **XML Hibernate Reverse Engineering (reveng.xml)**
- **New → Other → Hibernate → Hibernate Reverse Engineering File.** *Next* y nos pide ubicación, se debe indicar la carpeta **src** del proyecto.
- Otra vez *Next* y se visualiza una ventana desde donde indicaremos las tablas que queremos mapear. Primero elegimos nuestro **Console Configuration** que no es otro que el que hemos creado en el paso anterior y, a continuación, pulsamos en *Refresh* para que muestre la base de datos “ejemplo” y sus tablas.
- Seleccionamos tablas “depart” y “emple” y pulsamos el botón *Include*. Después *Finish*. (Observa siguiente imagen)
- Se abrirá el editor de Hibernate Reverse Engineering y en la pestaña **Source** podemos visualizar el XML generado



Hibernate C Package Exp

Hiper2
Proyecto1
src
hibernate.cfg.xml
JRE System Library [JavaSE-1.8]
MySQL Driver

Property	Value
Info	
derived	false
editable	true
last modified	6 de marzo de 2017, 0:53:17
linked	false

Proyecto1

Configure Table filters

Specify which catalog/schema/tables should be included or excluded from the reverse engineering.

Console configuration: Hiper2config

Database schema:

- ejemplo
 - depart
 - emple**
 - totales

Include...
Exclude...
Up
Down
Remove
Remove All

Refresh

Table filters:

!	Catalog	Schema	Table
	ejemplo	.*	depart
	ejemplo	.*	emple

?

< Back Next > Cancel Finish

Message	Plug-in	Date
Problems while reading database schema	org.hibernate.eclipse	6/3/17 1:47
org.jboss.tools.hibernate.runtime.spi	org.hibernate.eclipse	6/3/17 1:47

Quick Access

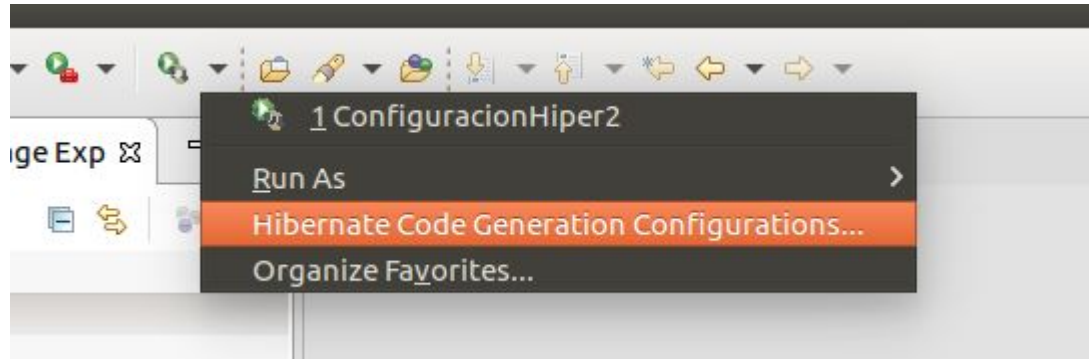
Query Outlin

No HQL editor open

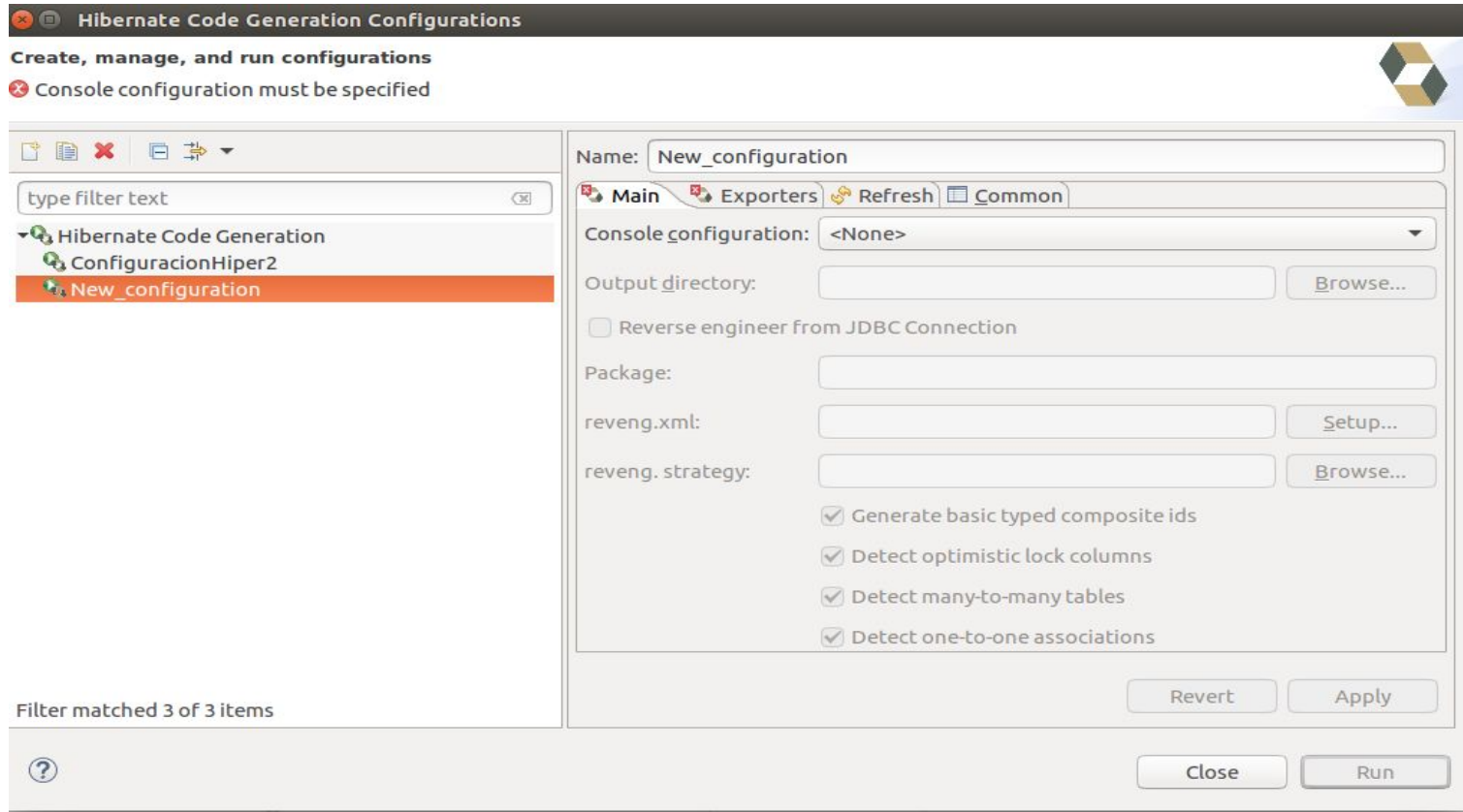


Generar las clases de la base de datos

- El siguiente paso es generar las clases de nuestra base de datos “ejemplo”.
- Para ello pulsamos en la flechita situada a la derecha del botón **Run As** y seleccionamos **Hibernate Code Generation Configurations**.

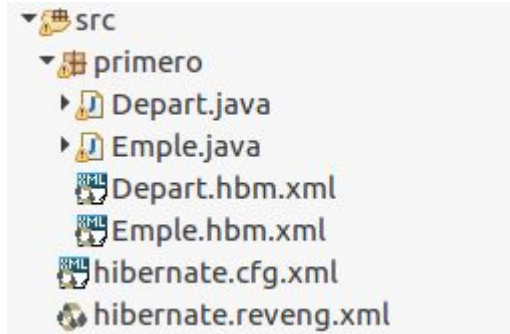


- En la nueva ventana hacemos doble clic en la opción **Hibernate Code Generation** del marco de la izquierda. Aparecerán varias pestañas:



- En la pestaña **Main** configuramos:
 - **Name** → Nombre para la configuración
 - **Console configuration** → Indicamos Proyecto1ConsoleConfiguration
 - **Output directory** → debe ser la carpeta **src**. Lo indicamos con el botón *Browse*
 - **Package** → escribimos el nombre del paquete donde se crearán las clases, por ejemplo “primero”
 - **revenge.xml** → localizamos el fichero revenge.xml creado anteriormente (*Setup*)
- En la pestaña **Exporters** se indica los fichero a generar. Se marcarán:
 - **Use Java 5 syntax**
 - **Domain code**
 - **Hibernate XML Mappings**
 - **Hibernate XML Configuration**
- Clicamos en *Apply* y posteriormente *Run*.

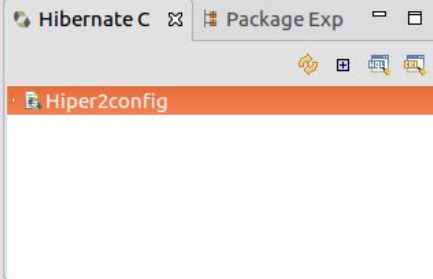
- Al ejecutarse nos genera un paquete llamado primero con las clases Java de las tablas emple y depart que contienen los métodos getters y setters de cada campo de la tabla. También contendrá los XML con información del mapeo de cada tabla.



- Conviene detenerse un poco y analizar cómo ha mapeado Hibernate cada tabla. Sobre todo es interesante la implementación de las claves ajenas.

Primera consulta en HQL

- Abrimos la perspectiva Hibernate. **Window** → **Perspective** → **Open Perspective** → **Other** → **Hibernate**.
- En la nueva pestaña, pulsamos en nuestra configuración del proyecto con el botón derecho y seleccionamos **HQL Editor**
- Aparece una nueva pestaña. Es el editor HQL. Podemos realizar consultas. Por ejemplo, si escribimos *from Emple* y pulsamos la flechita verde nos debe aparecer en la pestaña **Hibernate Query Result** el resultado de la consulta. (Ver siguiente imagen)
- Desde este entorno también se pueden realizar consultas SQL aunque hay ciertas restricciones. Por ejemplo, no se puede utilizar * en el SELECT



Property	Value
Identifier	
empNo	7369
Properties	
apellido	SÁNCHEZ
comision	
depart	
dir	7902
fechaAlt	1990-12-17
oficio	EMPLEADO
salario	1040

```
*Hiper2config
Hiper2config
1 from Emple
```

Name	Type	Value

Error Log	Hibernate Query Result	Hibernate Dynamic SQL Preview	Console
	primero.Emple primero.Emple@7F3c7f0c primero.Emple@21dd1f14 primero.Emple@5dc36b0 primero.Emple@606f5189 primero.Emple@2390148d primero.Emple@6afc9a54 primero.Emple@e71580d primero.Emple@1408a353 primero.Emple@371ff923 primero.Emple@393ffbae primero.Emple@3d0a6db4 primero.Emple@6cc9437d primero.Emple@24576926 primero.Emple@62f31489		

from Emple