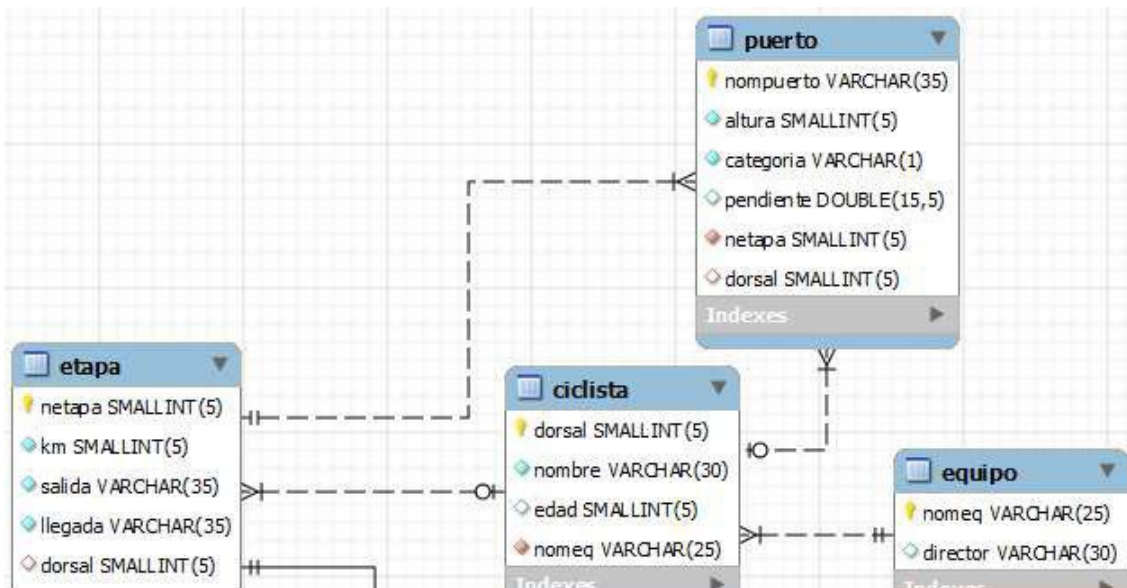


1. El siguiente Diagrama E-R representa la base de datos Ciclismo



Crea un proyecto Maven con el nombre p2\_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 p2\_hibernate
- Añade las dependencias necesarias para trabajar con Hibernate y MySQL
- Crea la clase HibernateUtil.java como se ha visto en clase
- Crea el fichero de configuración (hibernate.cfg.xml)
- Crea un paquete interfaz donde pondrás todas las soluciones para todos los ejercicios

a) Actividades a realizar sobre los pojos **Ciclista** y **Equipo**

- Crea los pojo Ciclista y Equipo, con constructores vacios, métodos getter y setter. Utiliza convenientemente las anotaciones @Entity y @id para que la clase sea mapeada con Hibernate.
- En el pojo Ciclista añade la relación “muchos a uno” de Ciclista a Equipo. En la clase Ciclista el atributo que referencia al equipo se llamará “equipo”.
- En el pojo Equipo añade la relación “uno a muchos” de Equipo a Ciclista. En la clase Equipo el atributo que referencia a los ciclistas será un ArrayList<Ciclista> y se llamará “ciclistas”.
- Añade al paquete interfaz un programa (clase **\_01a01DirectorDelCiclista**) que, dado el dorsal de un ciclista introducido por el usuario, muestre el nombre del director de dicho ciclista.
- Añade al paquete interfaz un programa (clase **\_01a02CiclistasDeEquipo**) que, dado un nombre de equipo introducido por el usuario, muestre el nombre los ciclistas de dicho equipo.

b) Actividades a realizar sobre el pojo **Etapas**

- Crea el pojo Etapas, con constructores vacios, métodos getter y setter. Utiliza convenientemente las anotaciones @Entity y @id para que la clase sea mapeada con Hibernate.
- Añade al pojo Etapas la relación “muchos a uno” de Etapas a Ciclista. En el pojo Etapas el atributo que referencia al ganador de la etapa se llamará “ganador”.
- En el pojo Ciclista añade la relación “uno a muchos” de Ciclista a Etapas. En la clase Ciclista el atributo que referencia a las etapas ganadas se llamará “etapasGanadas” y será un ArrayList<Etapas>
- Añade al paquete interfaz un programa (clase **\_01b01GanadorDeEtapas**) que, dado un número de etapa introducido por el usuario, muestre el nombre del ciclista ganador de dicha etapa.
- Añade al paquete interfaz un programa (clase **\_01b02EtapasGanadasPorCiclista**) que, dado el dorsal de un ciclista introducido por el usuario, muestre la ciudad de salida y de llegada de las etapas ganadas por dicho ciclista.
- Añade al paquete interfaz un programa (clase **\_01b03EtapasGanadasPorEquipo**) que, dado un nombre de equipo introducido por el usuario, muestre el nombre de sus ciclistas junto con el número de etapas que ha ganado cada uno de ellos.

c) Actividades a realizar sobre el pojo **Puerto**

- Crea el pojo Puerto, con constructores vacios, métodos getter y setter. Utiliza convenientemente las anotaciones @Entity y @id para que la clase sea mapeada con Hibernate.
- Añade al pojo Puerto la relación “muchos a uno” de Puerto a Ciclista. En el pojo Puerto el atributo que referencia al ganador del puerto se llamará

“ganador”.

- En el pojo Ciclista añade la relación “uno a muchos” de Ciclista a Puerto. En la clase Ciclista el atributo que referencia a los puertos ganados se llamará “puertosGanados” y será un ArrayList<Puerto>
- Añade al pojo Puerto la relación “muchos a uno” de Puerto a Etapa. En el pojo Puerto el atributo que referencia a la etapa se llamará “etapa”.
- En el pojo Etapa añade la relación “uno a muchos” de Etapa a Puerto. En la clase Etapa el atributo que referencia a los puertos que contiene se llamará “puertos” y será un ArrayList<Puerto>
- Añade al paquete interfaz un programa (clase **\_01c01GanadorDePuerto**) que, dado un nombre de puerto introducido por el usuario, muestre el nombre del ciclista ganador de dicho puerto.
- Añade al paquete interfaz un programa (clase **\_01c02PuertosGanadosPorCiclista**) que, dado el dorsal de un ciclista introducido por el usuario, muestre el nombre, la altura y la categoría de los puertos ganados por dicho ciclista.

2. Añade al proyecto los siguientes programas:

- a) (Clase **\_02aCrearEquipo**) Programa que solicite al usuario el nombre de un equipo y el nombre del director y añada un nuevo equipo con los datos indicados. ¿En caso del que el equipo ya exista, qué excepción se produce? Modifica el programa capturando la excepción para avisar al usuario en caso de que el equipo introducido ya exista.
- b) (Clase **\_02bModificarEquipo**) Programa que solicite al usuario el nombre de un equipo y el nombre del director y modifique el equipo indicado para que el director pase a ser el indicado. Si el equipo indicado no existe se avisará al usuario.
- c) (Clase **\_02cCambiarDeEquipo**) Programa que solicite al usuario el dorsal de un ciclista y el nombre de un equipo y cambie al ciclista al equipo indicado. El programa mostrará mensajes de error en caso de que el ciclista indicado no exista, en caso de que el equipo indicado no exista y en caso de que el equipo indicado sea el mismo al que pertenece ya el ciclista.
- d) (Clase **\_02dCrearCiclista**) Programa que solicite al usuario el dorsal, nombre y nacimiento de un ciclista y el nombre de un equipo. El programa creará un nuevo ciclista con los datos indicados. El ciclista pertenecerá al equipo indicado. Se mostrarán mensajes de error en caso de que el ciclista ya exista y en caso de que el equipo no exista. (Para facilitarte las pruebas crea el ciclista en el equipo que has creado en el paso \_02a)

3. (Clase **\_03EliminarEquipo**) Escribe un programa que elimine el equipo que indique el usuario y todos sus ciclistas. (Para facilitarte las pruebas elimina el equipo que creaste en el paso \_02a)
4. (Clase **\_04EliminarEquipoYCambiarCiclistas**) Escribe un programa que elimine el equipo que indique el usuario pasando previamente sus ciclistas a otro equipo (que también indicará el usuario)
5. Crea una clase llamada **\_05HQLBasicas** para ejecutar las siguientes consultas HQL
  - a) Ciclistas nacidos entre el 1/1/1979 y 31/12/1980.

Revisa cómo se guardan las fechas en la base de datos

- b) Número total de ciclistas
  - c) Número total de ciclistas del equipo Banesto
6. Crea una clase llamada **\_06HQLVariasTablas** para ejecutar las siguientes consultas HQL. Ejecútalas previamente en el HQL Editor
  - a) De cada etapa, ciudad de salida y llegada y nombre del ciclista que la ha ganado
  - b) De cada equipo, nombre, director y número de ciclistas que tiene.