HYBERNATE

No hay queries, es mejor, se llama HQL

Mapea solo los objetos con las tablas

Dependencias de mysql, hibernate, el validator, el glassfish de jararta y el resto

Plugin de intellij de hibernate enabled

En la clase JPAUtil llamas al persistence.xml que esta en resources que es como el properties de antes

Hay 3 maneras de pedir datos a base de datos, con sql, con hql a pelo, o con hql en una constante con la anotación nativa de hibernate en el model, que es como lo tenemos que hacer.

Si no metes @column en un atributo de un objeto del model no te lo devuelve el hibernate, si ese atributo es un objeto deberas poner @Transient

El get de un id como PK se hace con un entity.find(ObjetoQueDevuelve,id)

Siempre que modificas la base de datos tienes que hacer una transacción

Las excepciones se throwean de normal, luego haces un e.getcause instanceof “tuexcepciondehybernate” y la sacas, porque se hace asi ahora, también puedes hacer un catch de PersistenceException en vez de Exception

??que es cascade persist (solo para add)

Cuando tienes un onetomany en la otra clase debes tener un manytoone, onetomany usa muchos recursos para que encuentre la fk a la que se refiere, por lo que no lo uses de mas o suspenso en el examen. Si en la lista de un objeto que hay relación 1 n ese objeto de la lista (n) tiene un id de referencia a su relación (1) y luego aparte esa relación tiene una lista declarada (lista de n), hybernate entra como en una especie de bucle al mostrar los datos, por que tienes que usar la notación @ToString.Exclude, también en la lista debes cambiar la manera de la que pide la lista de N con fetch= FetchType.EAGER o pedir un .size para que el vago por defecto detecte que se necesite la lista y lo cargue