[Como funciona hibernate.hbm2ddl.auto?](https://pt.stackoverflow.com/questions/90723/como-funciona-hibernate-hbm2ddl-auto)

Quais são os values que posso utilizar nessa propiedade? ex: Update

<prop key="hibernate.hbm2ddl.auto">update</prop>

Como funciona? Quando devo utilizar? é uma boa prática?

1º Resposta:

As opções são essas:

* **validate**: validar o schema, não faz mudanças no banco de dados.
* **update**: faz update o schema.
* **create**: cria o schema, destruindo dados anteriores.
* **create-drop**: drop o schema quando ao terminar a sessão.

Ai você tem que avaliar o que é melhor para seu projeto, geralmente eu utilizo o update.

Link da resposta no SOen [link](https://stackoverflow.com/a/1689769/5165064)

# [What are the possible values of the Hibernate hbm2ddl.auto configuration and what do they do](https://stackoverflow.com/questions/438146/what-are-the-possible-values-of-the-hibernate-hbm2ddl-auto-configuration-and-wha)

From the [community documentation](https://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html" \l "configurations-hbmddl):

hibernate.hbm2ddl.auto Automatically validates or exports schema DDL to the database when the SessionFactory is created. With create-drop, the database schema will be dropped when the SessionFactory is closed explicitly.

e.g. validate | update | create | create-drop

So the list of possible options are,

* *validate*: validate the schema, makes no changes to the database.
* *update*: update the schema.
* *create*: creates the schema, destroying previous data.
* *create-drop*: drop the schema when the SessionFactory is closed explicitly, typically when the application is stopped.
* *none*: does nothing with the schema, makes no changes to the database

These options seem intended to be developers tools and not to facilitate any production level databases, you may want to have a look at the following question; [Hibernate: hbm2ddl.auto=update in production?](https://stackoverflow.com/questions/221379/hibernate-hbm2ddl-autoupdate-in-production)

2º Resposta:

### **Como funciona?**

Respondida pelo nosso amigo [Rafael](https://pt.stackoverflow.com/users/20706/rafael).

### **Quando devo utilizar?**

Isso é um pouco ambíguo, pois pode desencadear boas discussões, mas é bom pensar nisso como dar responsabilidade a uma tecnologia que teoricamente vai cuidar de todas as ações de desenvolvimento no database.

*Será que é uma boa idéia deixar acontecer de forma automática, a criação, atualização ou remoção de qualquer entidade pelo Hibernate?*

Não, por mais que o todo o desenvolvimento para o database forme o mesmo codebase, não significa que a automatização de um sobre o outro irá funcionar. Validar seus scripts de uma forma manual é sempre uma forma mais segura para o desenvolvimento. Nunca utilize isso em produção, pode ser um erro fatal.

Mas podemos falar em contexto, se o seu contexto é fazer um projeto próprio de estudo e teste, talvez sim, mas eu não acredito que seja uma boa prática. Em minha experiência com ferramentas que costumam fazer algo relacionado a desenvolvimento de forma automática, tendem a falhar miseravelmente.

### **é uma boa prática?**

Não. Vejo como uma péssima prática, tanto do lado do desenvolvedor quanto do lado do cliente que vai utilizar o produto. Dar responsabilidades de desenvolvimento a qualquer ferramenta não é uma boa prática. Boa prática é você ter total controle do produto, desde a forma como é entregue, até a sua arquitetura e code quality. Pense o quão simples é para o hibernate replicar uma linha de código errada sua para o database e dessa forma causar o caos e isso esteve totalmente fora do seu alcance.

Mas como falei, podemos gerar boas discussões, pois se formos focar em boas práticas não utilizaríamos ORM, pois é com certeza um anti-pattern para Orientação a Objetos.