**hbm2ddl in Hibernate**

hbm2ddl stands for **Hibernate Mapping to Data Definition Language**. It is a Hibernate feature used to automatically generate database schema (DDL statements like CREATE, DROP, ALTER) from your Java entity classes.

**🛠️ Where is hbm2ddl used?**

It's used in the **Hibernate configuration file** or **persistence unit settings** to manage the **automatic schema generation**.

**🔧 Configuration Example:**

In hibernate.cfg.xml:

<hibernate-configuration>

<session-factory>

<!-- Other properties -->

<property name="hibernate.hbm2ddl.auto">update</property>

</session-factory>

</hibernate-configuration>

In application.properties (Spring Boot):

spring.jpa.hibernate.ddl-auto=update

**📘 Available Values for hbm2ddl.auto:**

| **Value** | **Description** |
| --- | --- |
| none | (default) No action will be performed. |
| validate | Validates schema, makes sure it matches entities. Doesn't modify DB. |
| update | Updates the schema (adds new columns, tables, but won’t delete existing ones). |
| create | Drops the existing schema and creates a new one every time. |
| create-drop | Same as create, but drops the schema when SessionFactory is closed. |
| drop | Drops the schema on startup. Rarely used. |

**✅ Example Usage:**

**Java Entity:**

@Entity

public class Student {

@Id

@GeneratedValue

private int id;

private String name;

}

**With hibernate.hbm2ddl.auto=update**, Hibernate will generate:

create table Student (

id int auto\_increment,

name varchar(255),

primary key (id)

);

**🚨 Important Notes:**

* **create and create-drop** can **delete all your existing data** — use only in development.
* Use validate in **production** to ensure DB and entities match without risk of data loss.
* update is convenient in development but risky in production as it may fail silently or cause unexpected issues.