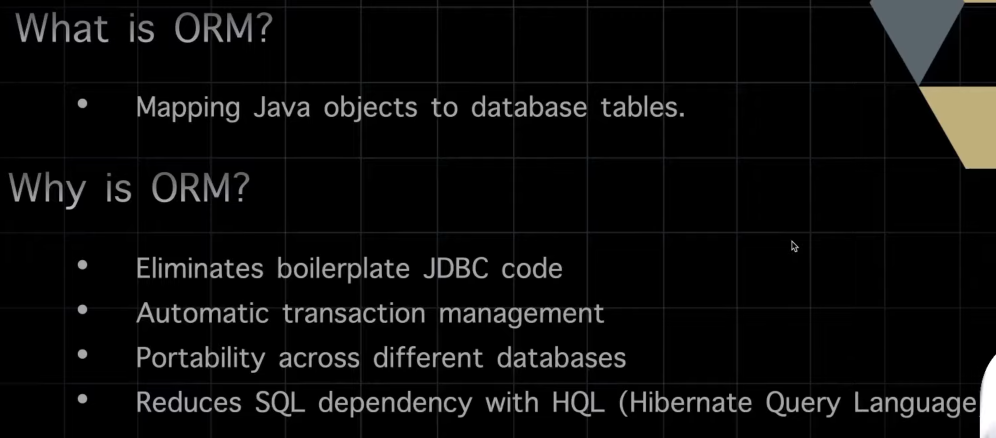
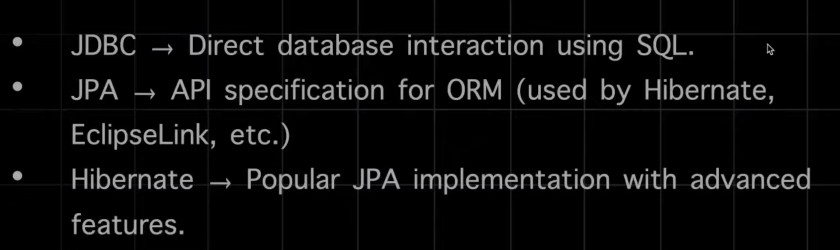


JDBC is used for interaction between app and DB. Using JDBC we have to write too much sql queries and other optimizations. (A lot manual work)

ORM solves this problem. It helps in saving Java class object to db table by defining mappings in config file. We give this config file to ORM tool. Allow automatic db operations.

In jdbc we directly write query as per our DB. So application becomes tightly coupled with that particular DB. In future changing db will need changing queires. ORM helps us to write db independent query. We use JPQL or HQL.

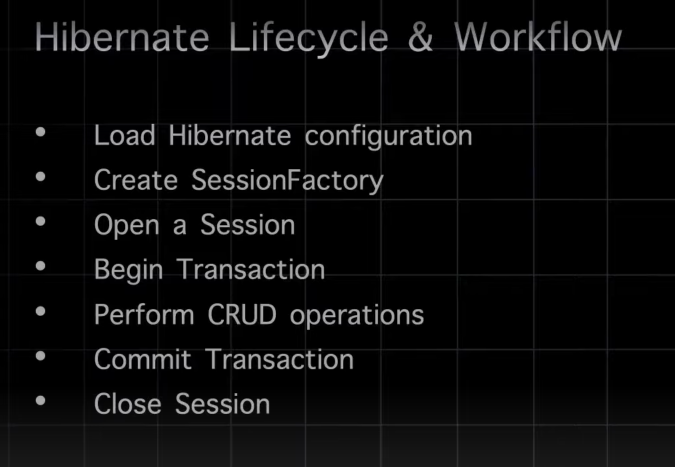


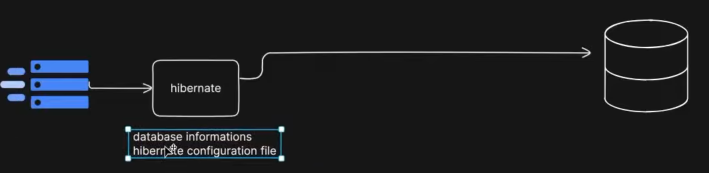


SessionFactory is single for whole application. Using it we can create sessions and perform db operations.

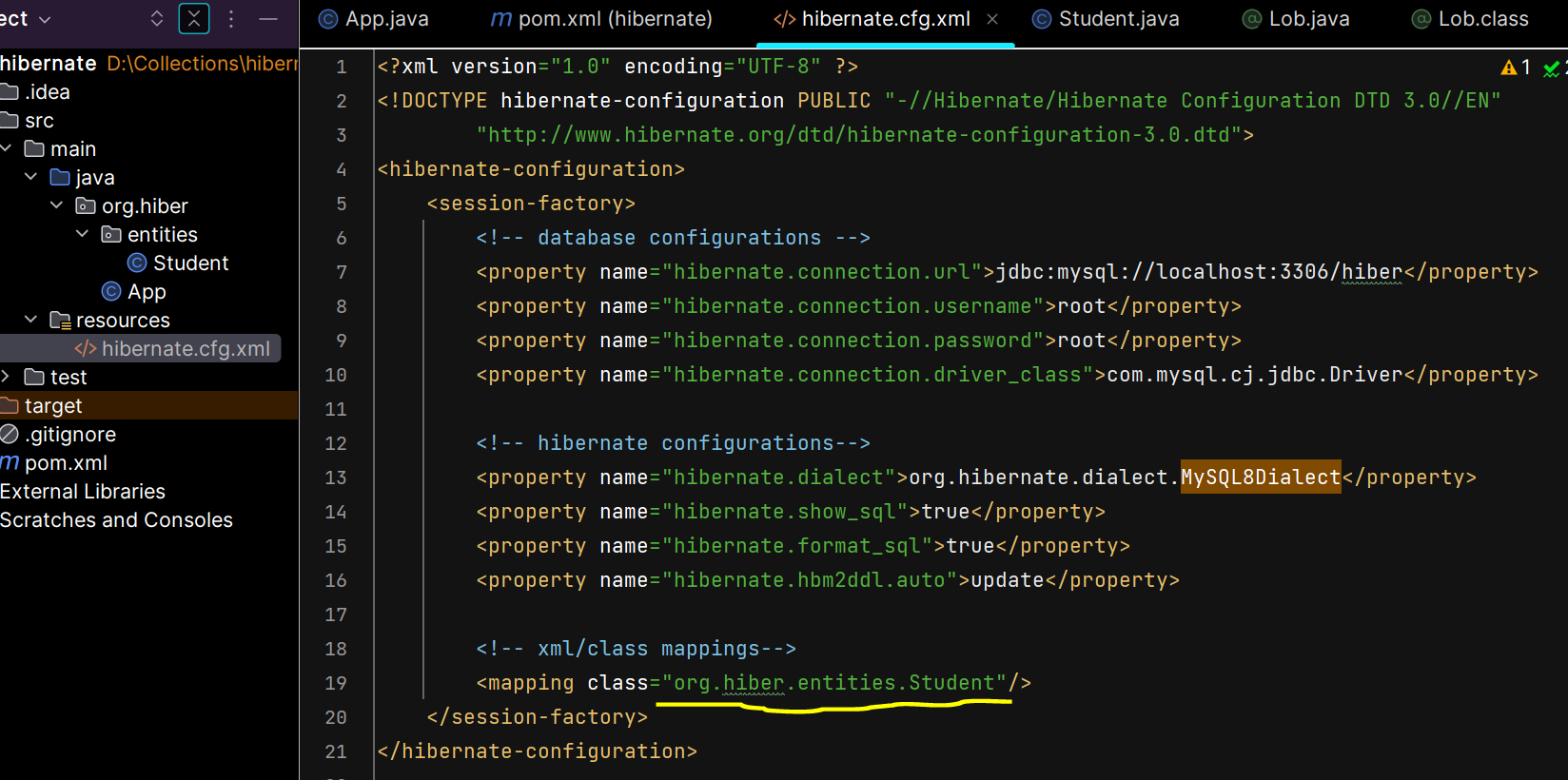
Using Session u can open a Transaction and commit/rollback.

Hibernate.cfg.xml - This file contains important configuration settings that tell Hibernate how to connect to a database, manage entities, and handle other Hibernate-related functionalities.











**Notes :-**

1. If you don’t use @Table, Hibernate will assume the default naming strategy and map the entity to a table with the same name as the class
2. @Column is also only required when you want to specify or customize the attributes of the column mapping.
3. @Lob used when we have to store large value in db.
4. <property name="hibernate.hbm2ddl.auto">update</property> this property creates the table if not already present.



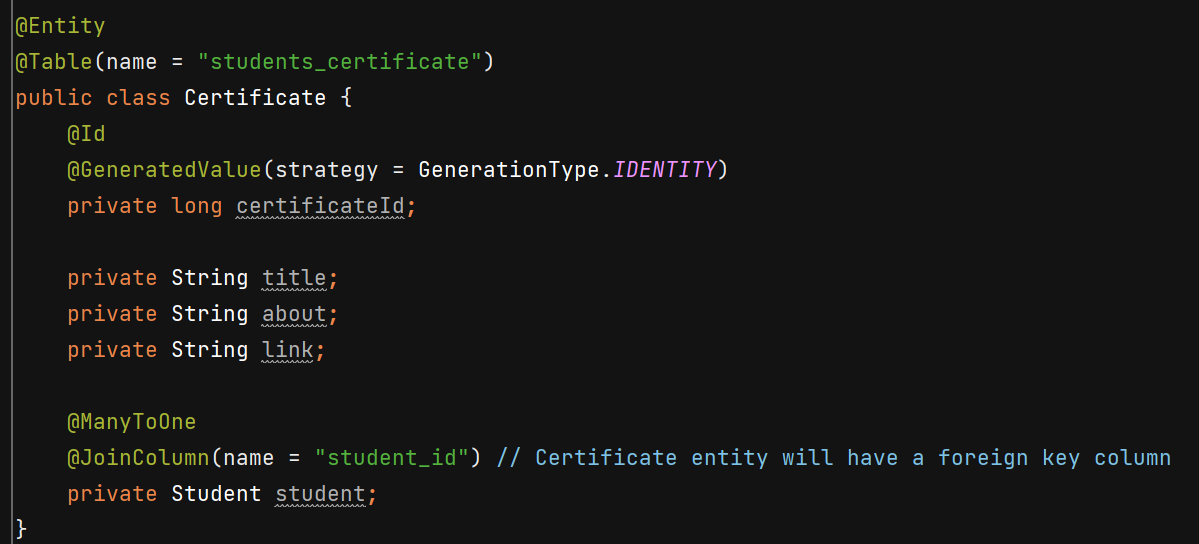


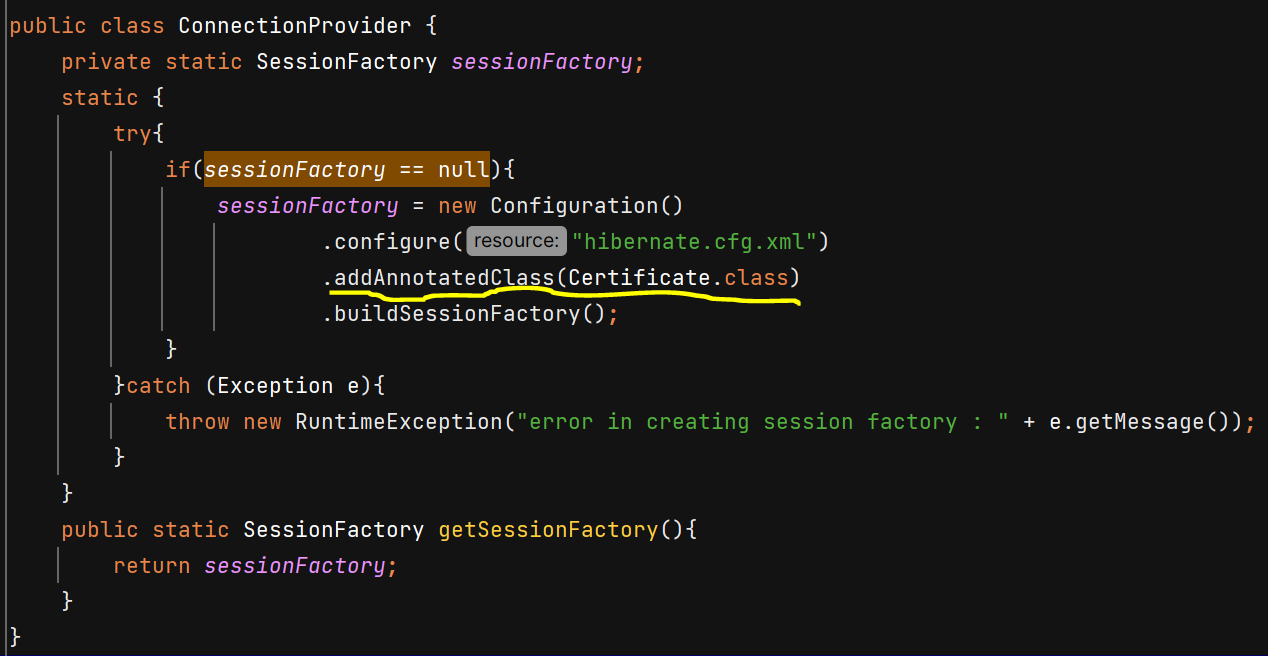
**Making One-To-Many relationship between Student & Certificate**

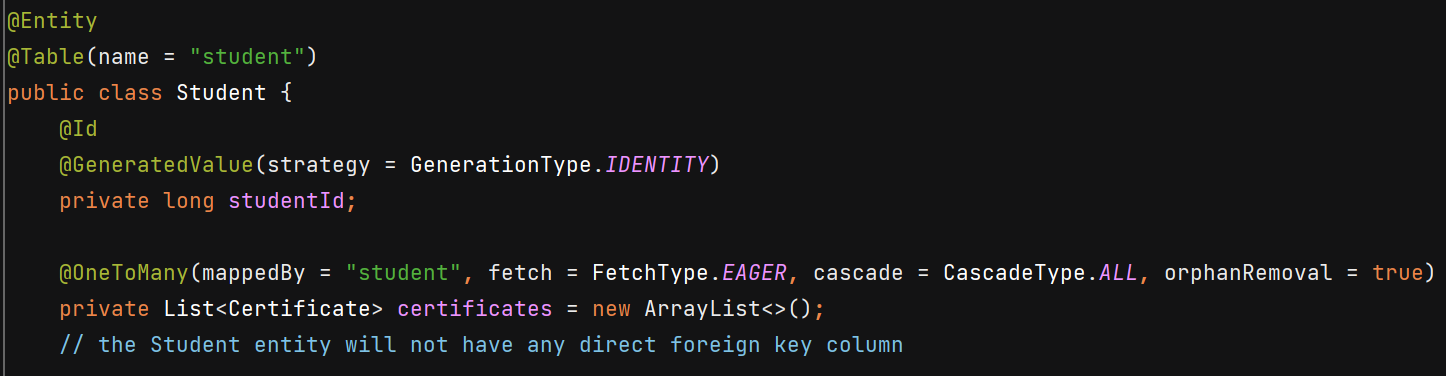
mappedBy = "student" => refers to the field student in the Certificate class

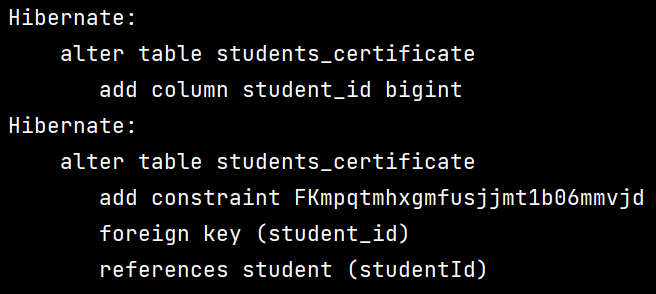
cascade = CascadeType.*ALL* => **all operations** performed on the parent entity will be **propagated** to the associated child entities

orphanRemoval = true => orphaned entities are automatically deleted when they are no longer associated with their parent entity. Ex - if a Student no longer has a Certificate in their list of certificates, that certificate will be **automatically deleted.**









**CRUD Operations**

session.get() method in Hibernate finds an entity using its primary key.

