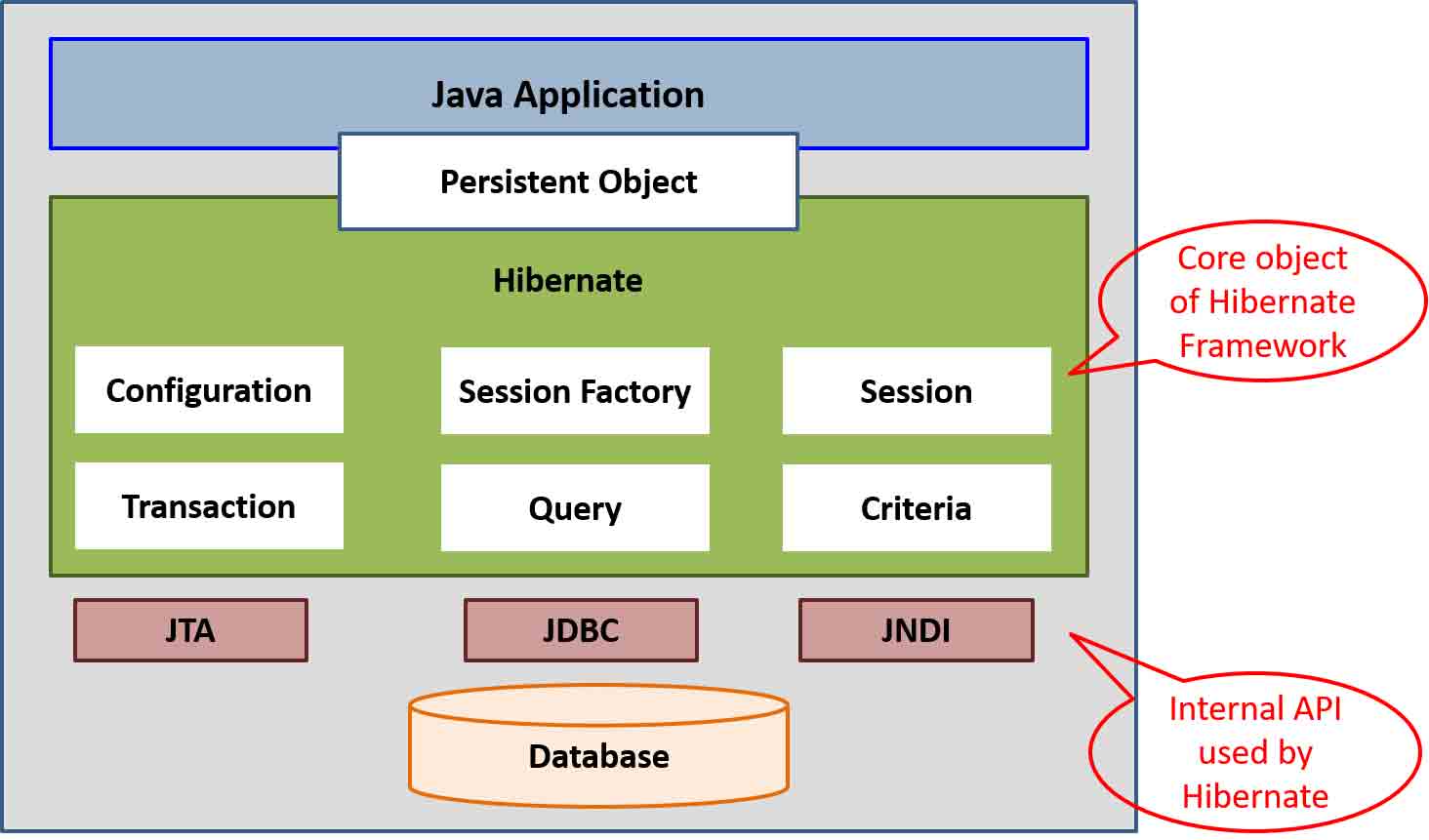
**Hibernate**

**What is Hibernate?**

Hibernate is Java framework that help to interact with database.

It is ORM tool.

Is use to perform persistence operation on Database with less or no database queries.

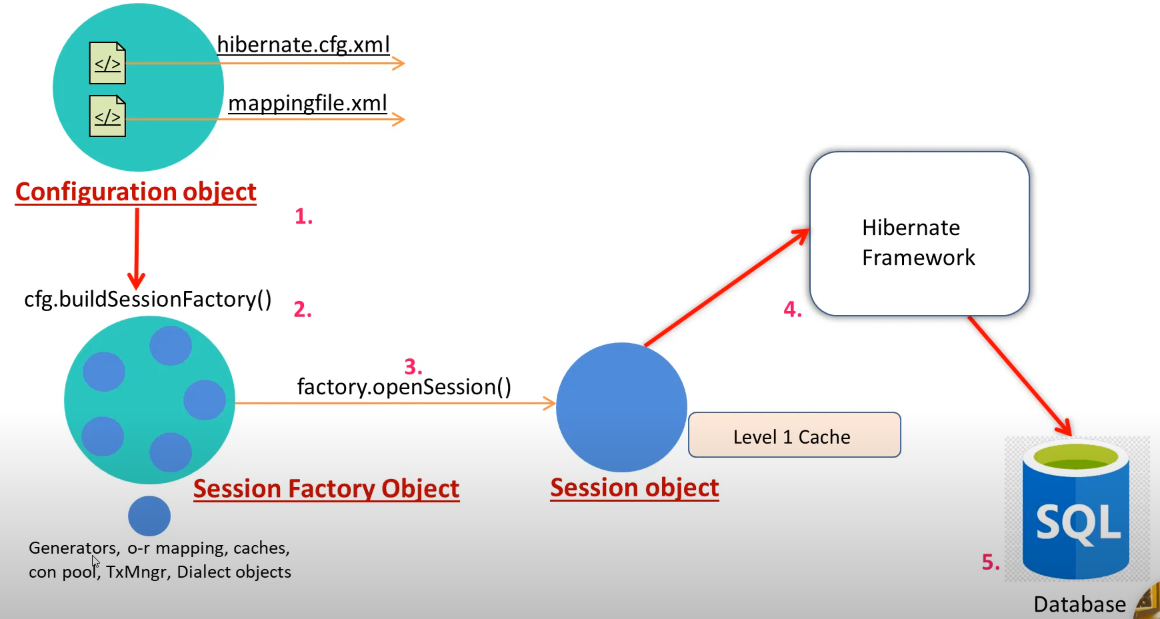


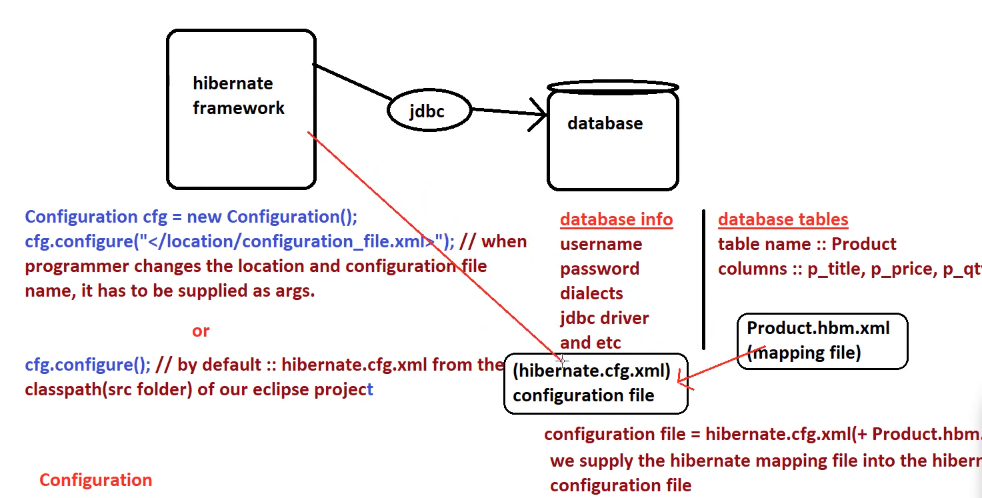
What is DAO Data Access Object Class?

DAO is related to database which we will use for modification or updating in database and make them reusable.

3 important Objects of Hibernate Persistence Logic.

1. Configuration Object – class
2. Session Factory Object – Interface – org.hibernate
3. Session Object





**Create Hibernate Simple Project.**

1. Create Maven quick start project
2. Add hibernate dependency in pom.xml file

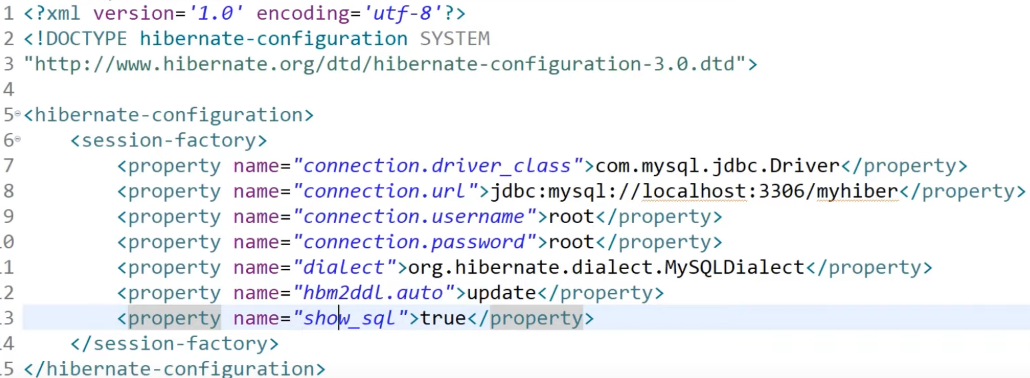
<dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>7.0.0.Alpha2</version>  
 <type>pom</type>  
</dependency>

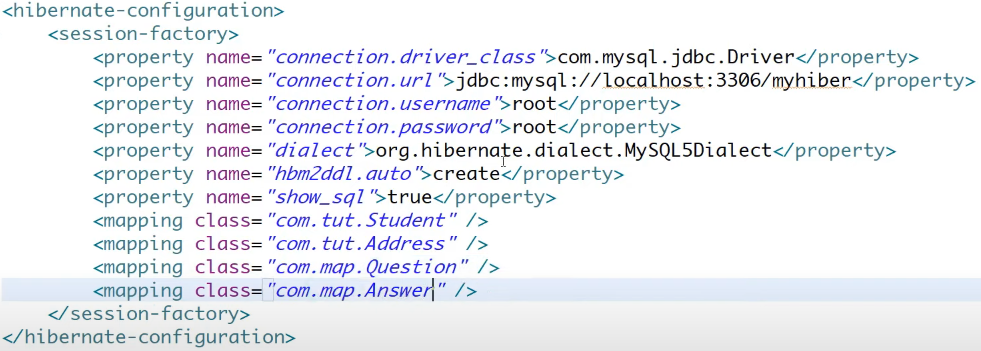
1. Create hibernate.cfg.xml file in src/main/java.
2. Search hibernate dtd on google and copy this
3. Write hibernate configuration for JDBC connection in this file.

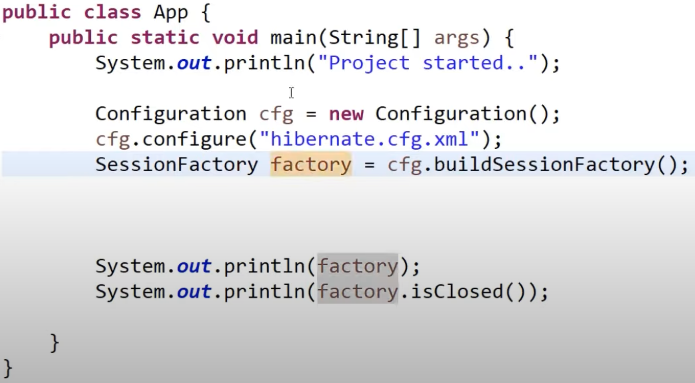
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  
<hibernate-configuration>  
 <session-factory>  
  
 <property name="connection.driver\_class">com.mysql.jdbc.Driver</property>  
 <property name="connection.url">jdbc:mysql://localhost:3306/hibernate</property>  
 <property name="connection.username">root</property>  
 <property name="connection.password">root</property>  
 <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>  
 <property name="hbm2ddl.auto">update</property>  
 <property name="show\_sql">true</property>  
  
 </session-factory>  
</hibernate-configuration>

1. Now in main class file we have to create SessionFactory interface variable.

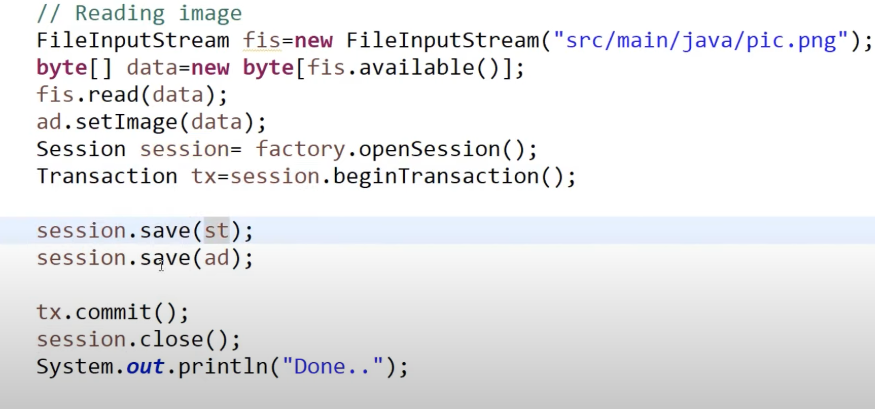
public class App   
{  
 public static void main( String[] args )  
 {  
 Configuration cfg = new Configuration();  
 cfg.configure("hibernate.cfg.xml");  
 SessionFactory factory = cfg.buildSessionFactory();  
 factory.close();  
 }  
  
}











**Fatching Object**

Session provied two method for fetch data using hibernate form database.

**Get()**

Use when we not sure that object is present or not.

It will return null if not present.

**Load()**

Use when we sure the object is present.

It will return ObjectNotFoundException if not present.

**Embeddable Object @Embeddable**

It will tell to create the reference object fields with entity class.

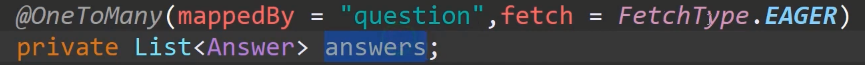
For example if we have Student class and in this class we have id, name, address

Now address is other class. We have to annoted class as @Embeddable

**Fetch Type in Hibernate.**

Eager Loading – it will load data on query running.

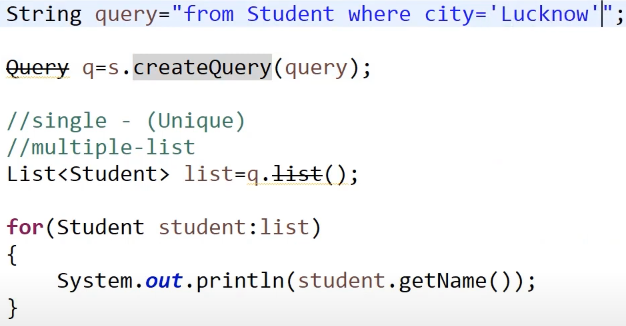
Lazy Loading – it will load data only when we call getter() or size() method explicitly.

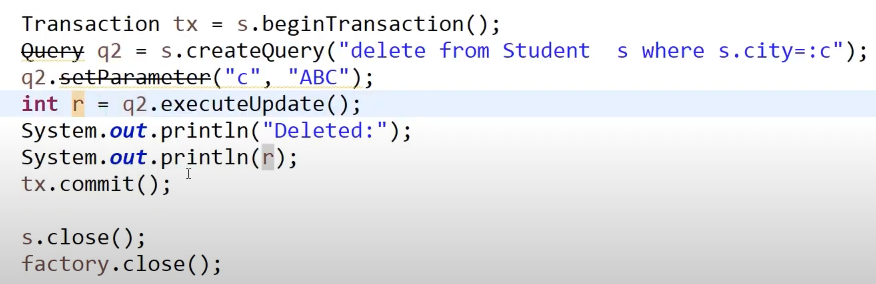


**Hibernate Object Life-cycle States**

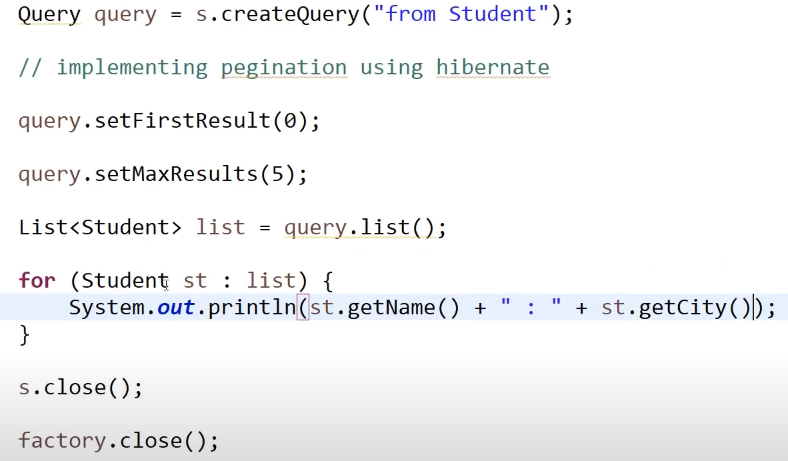
* Transient State – when we create object.
* Persistent State – when we save this object mean it will associated with database and session.
* Detached State – It mean we clear or close the session.
* Removed State – When we call delete. It means remove data from database but it is associated with session.

**Hibernate Query Language HQL**





**Pagination in Hibernate**



**Cascading in Hibernate**

It means it will store or save the data which is reference of other class implicitly.

We need to just add cascade attribute in this class field.

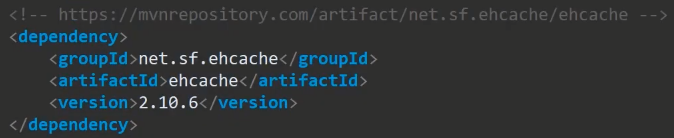


**Caching in Hibernate**

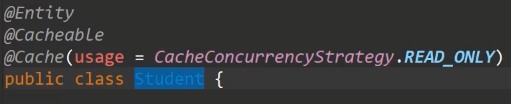
* Caching is the mechanism to enhance the performance of application.
* Caching is use to reduce the number of queries.
* Type of Caching

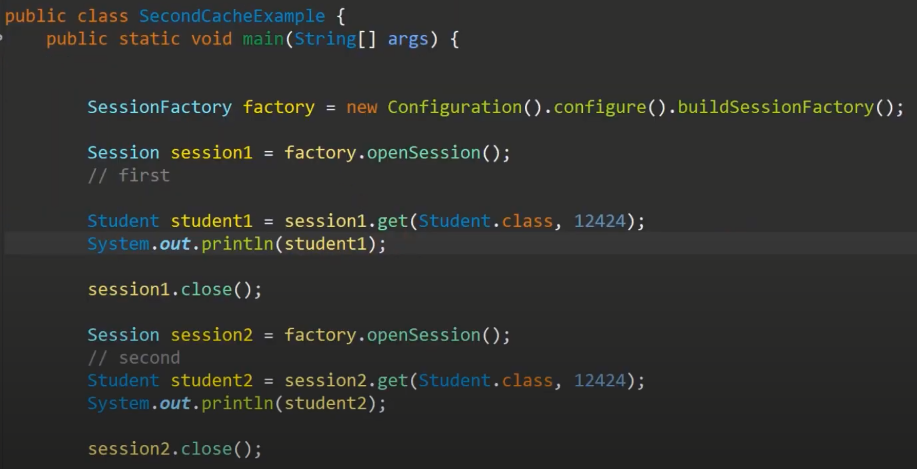
First Level – associated with Session Object – By Default

Second Level – associated with SessionFactory – Explicitly enable configuration









**Hibernate Interview Questions**

**What is Hibernate and why to use?**

Hibernate is ORM Object Relational mapping tool which map the Java object to database tables.

It Provide us to JPA implementation to achieve this mapping with XML configuration or JPA Annotation.

**Important Interface in Hibernate?**

**SessionFactory** – org.hibernate.SessionFactory

It is an interface which is use to retrieve the session object.

It’s like one SessionFactory per database connection.

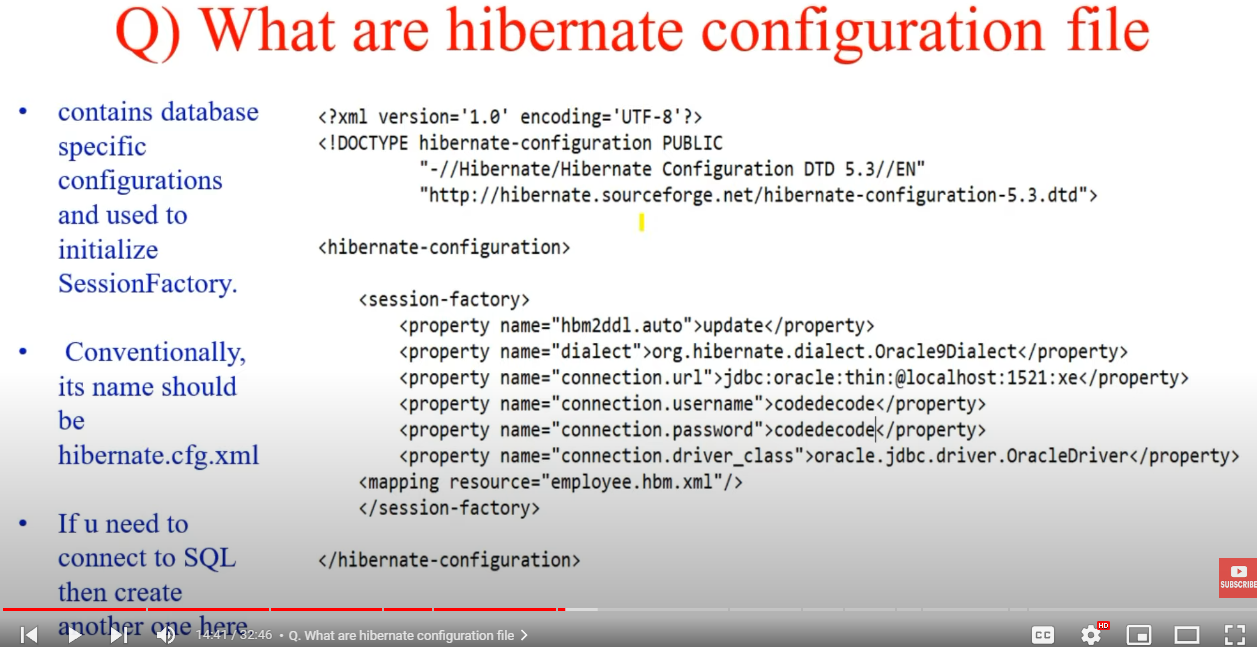
**Session** – org.hibernate.Session

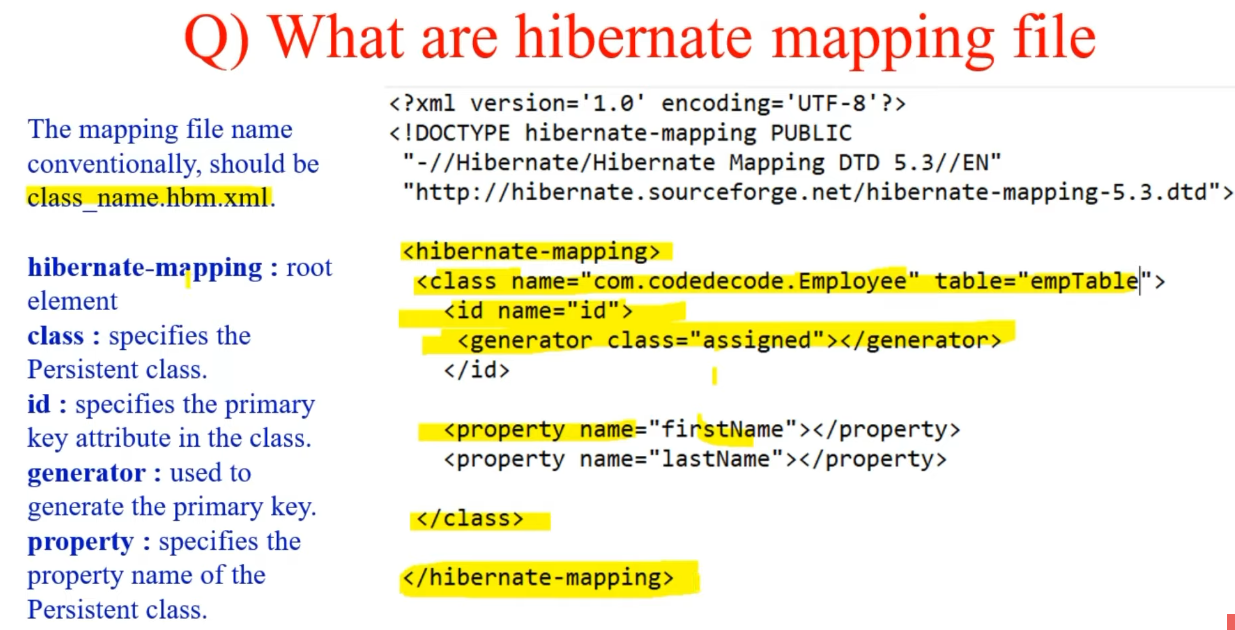
Its factory for Transaction. It use to get physical connection with database.

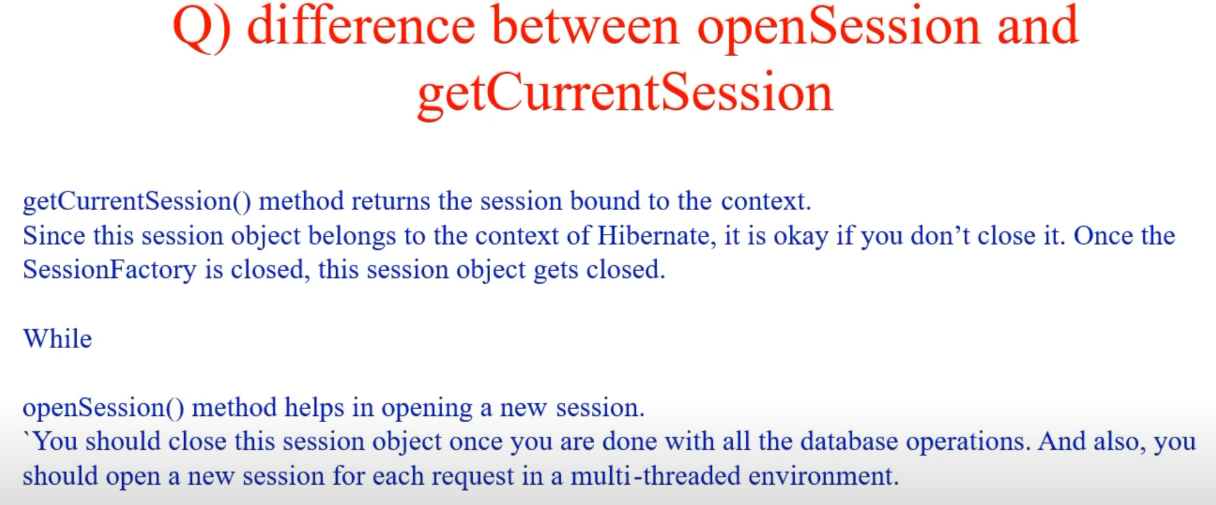
It provide the CRUD method operations.

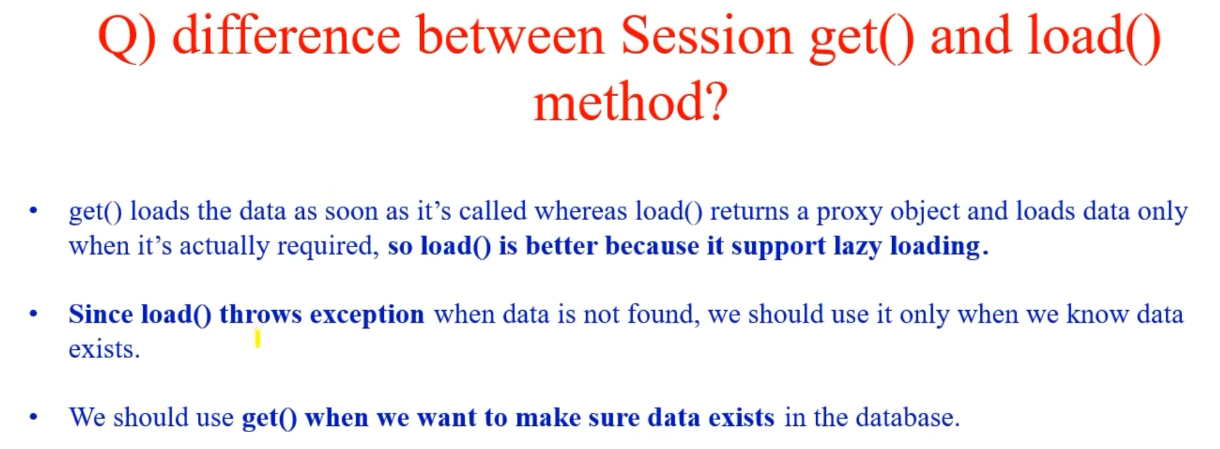
**Transaction** - org.hibernate.Transaction

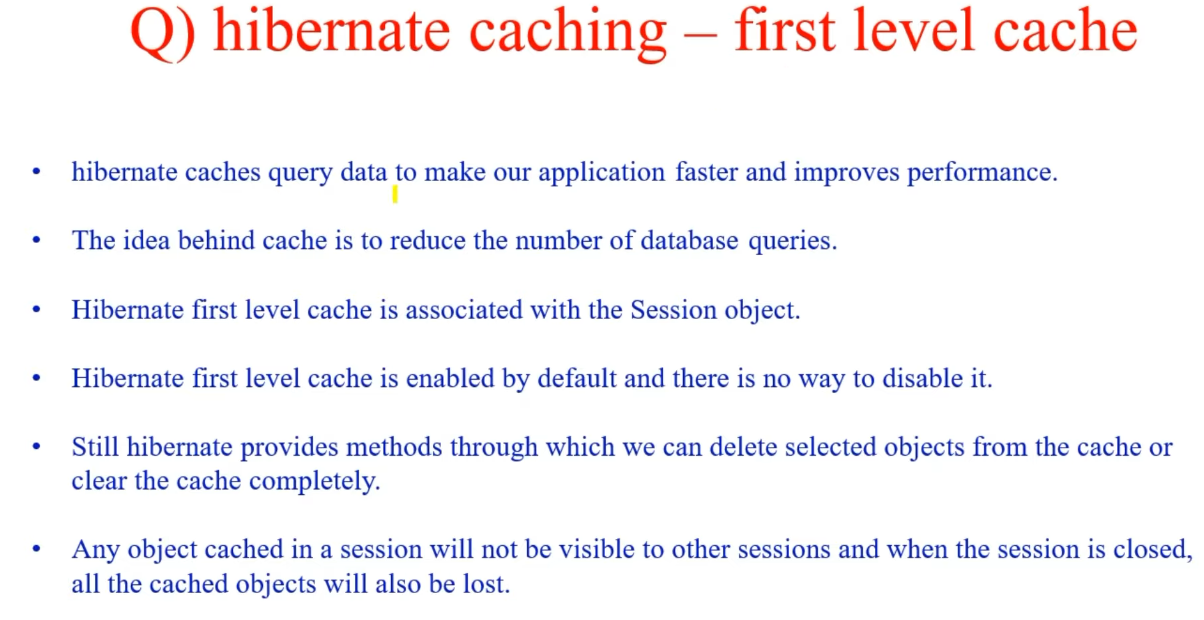
It will begin for CRUD operation process.

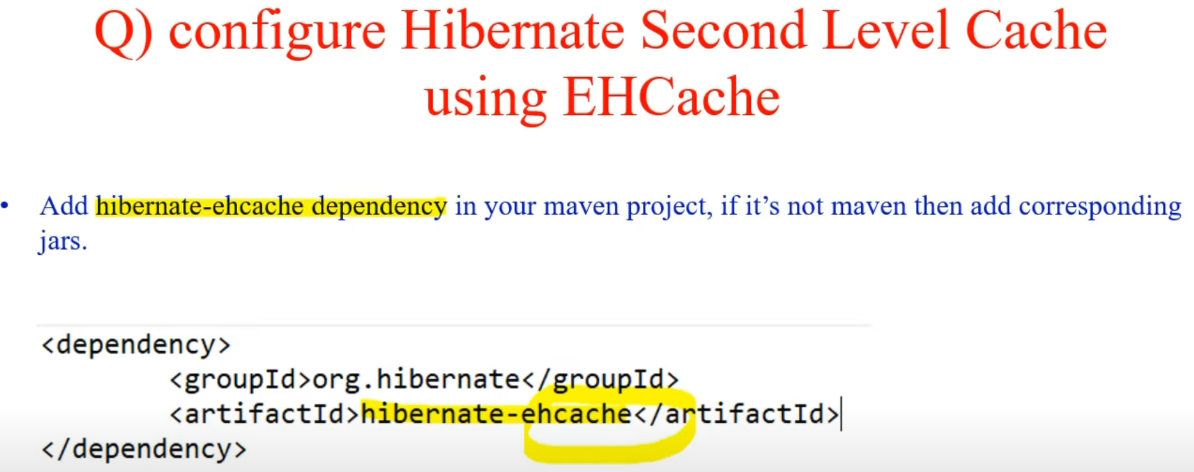


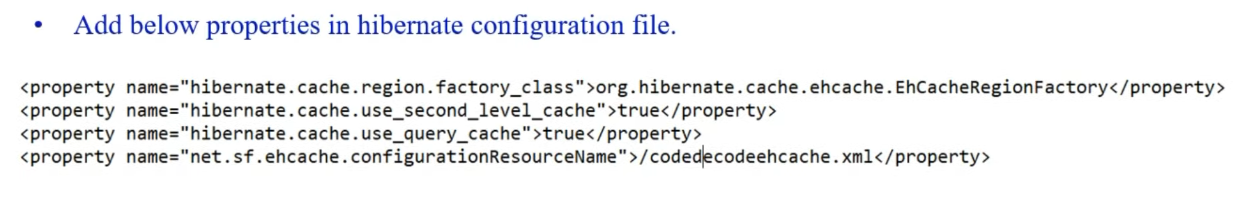


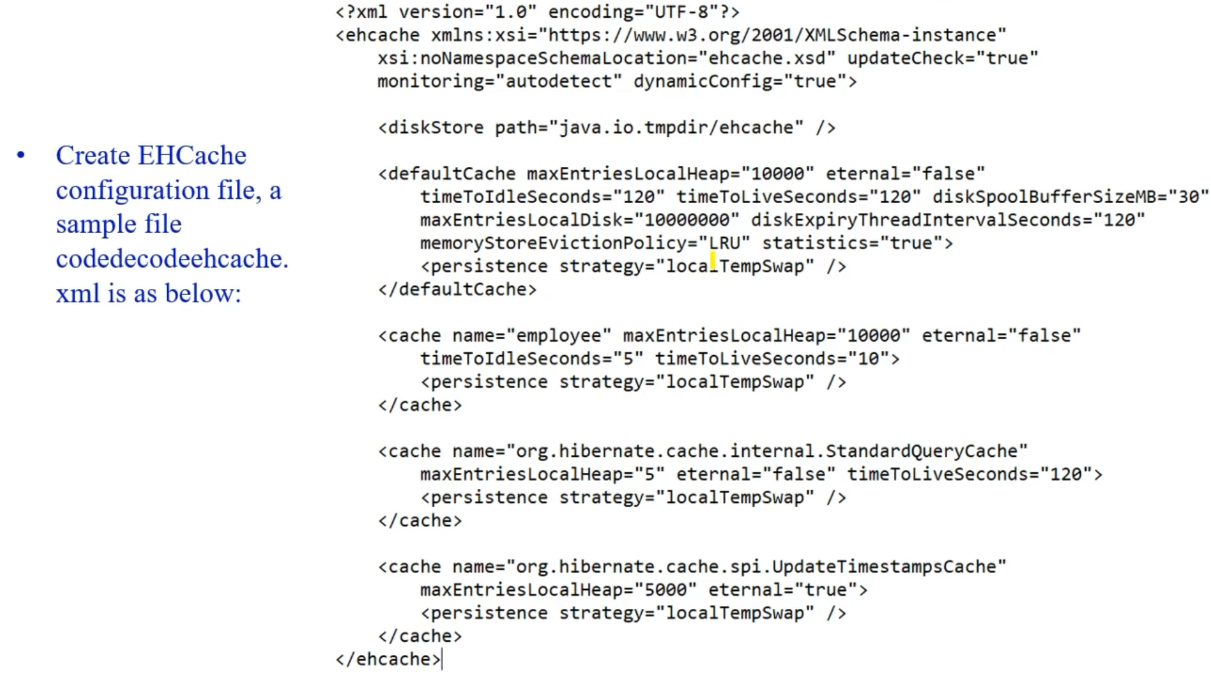


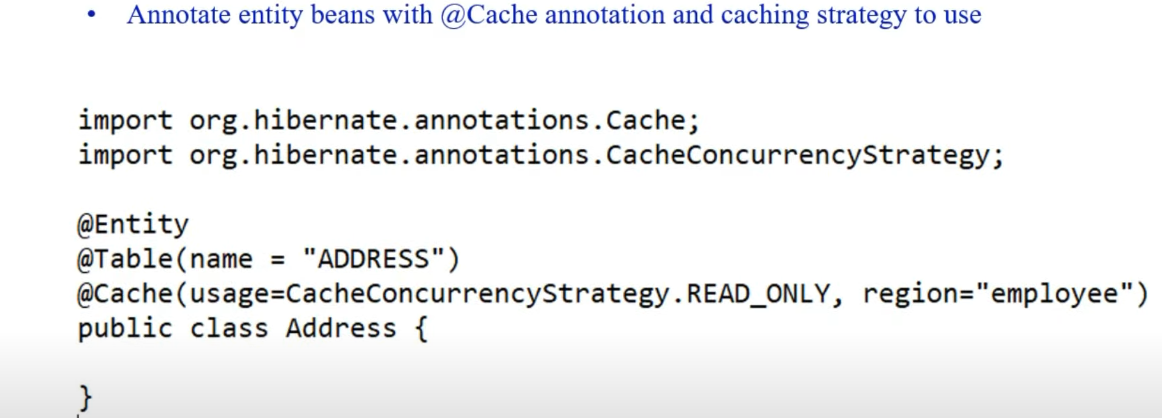












**How to Ingrate Hibernate with Spring?**

We need to add dependency in pom.xml file. (Hibernate , JPA, Core Hibernate, MySql).

Create entity class, Controller class, Service class and DAO layer.