

Hibernate is a ORM = Object Relational Mapping

hibernate.cfg.xml ---

What all things hibernate is doing for us ?

1. Connectivity
2. Creates a pool of connections - pre created connections will improve response time
3. We provide mapping classes to hibernate --- hibernate creates tables ( DDL commands )
4. Using session API --- hibernate will persist the object and it will do the DML commands

You deal with objects , I deal with database !!!

HQL = **hibernate query language** will help us write queries using classes and its properties.

SQL = we write queries using tables and their columns

APIs for basic DML commands ---

insert	session.save() , session.saveOrUpdate()	
update	add the object is persistent state ( we used find ) and then call setter methods	obj = session.find(StudentEntity.class, primary key); obj.setName("new Name");
Delete	session.delete()	
select a record by primary key	session.find( class , primary key value )	
select multiple records	Query q = createQuery( <b>hql</b> ) Arraylist<Entity> list = q.list();  Query q = createNativeQuery( <b>sql</b> ) q.list()	obj = session.find(StudentEntity.class, primary key);  session.delete(obj);

Establishing table relationships through Hibernate!!!!

Relationships Mapping -----

1. One to One ---- both tables foreign key
2. One to Many ---- employee did not have fk , but acct had fk
3. Many to Many --- third table

1. DB Relationships can be established through hibernate
2. This can be done use annotations like --- @OneToOne , @OneToMany , @ManyToOne , @ManyToMany  
@JoinColumn , @JoinTable , @mappedBy INSIDE the Entity classes !!!!!

3. All the FK etc are generated properly by the hibernate
4. if we insert one entity the dependent entities are inserted/updated /selected /deleted --

CASCADE

HW ----

Create different core java projects --for each relationship ---build path, hibernate.cfg.xml , main

1. Onetooneex
  - study.entity.Employee,
  - study.entity.Address ,
  - prudent.study.HibernateApplication

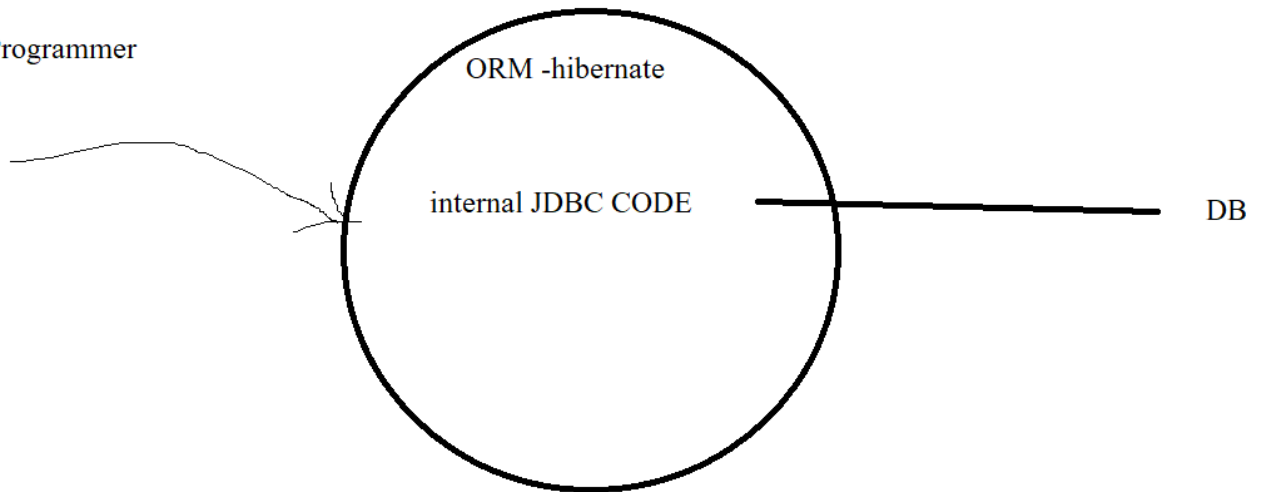
2. Onetomanyex -----
  - hibernate.test.oneToMany.foreignKeyAsso.EmployeeEntity ,
  - hibernate.test.oneToMany.foreignKeyAsso.EmployeeEntity .AccountEntity ,
  - hibernate.test.oneToMany.TestForeignKeyAssociation

3. manytomanyex

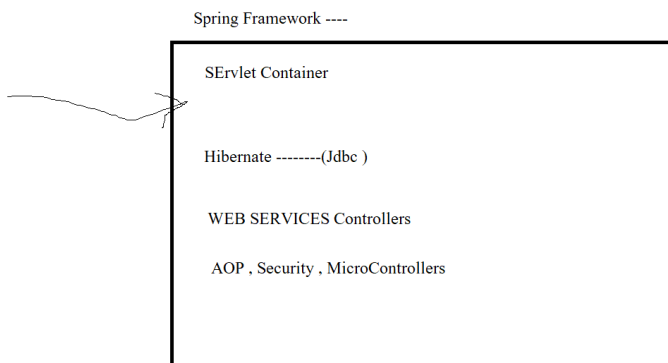
---hibernate.test.manyToMany.joinTable.ReaderEntity

----hibernate.test.manyToMany.joinTable.SubscriptionEntity  
-----hibernate.test.manyToMany.TestJoinTable

Java Programmer



Spring Framework !!!! ----- This container is WRAPPING many containers !!!



Till Now ----

IDE ---- eclipse for JEE !!!!

We will use Eclipse based IDE for Spring Development ---- STS = Spring Tool Suite

<https://spring.io/tools> } scroll down ---

## Spring Tools 4 for Eclipse

windows download !!!!!

Spring FRAMEwork ---

Manage Life Cycle of Components called as BEANS !!!!!

Servlet container was managing life cycle of Servlet

Hibernate Container was managing life cycle of Entity

Spring Container is managing the life cycle BEANS !!!!!

Beans are POJO classes !!!

**Dependency INJECTION** of properties is done by Spring Framework !!!

JARS !!!! Management ---- MAVEN !!! ----- Go through what is maven !!!!!

---