



Ø What is Hibernate?

- **Hibernate is a java framework that simplifies the development of java application to interact with database.**
- **Hibernate is a ORM (Object Relational Mapping) tool.**
- **Hibernate is an Open source, light weight.**
- **It is invented by Gavin King in 2001.**
- **Any type of application can build with Hibernate Framework.**

Ø How create a maven project?

- **In Eclipse first we have to go
fileànewàprojectàmavenàcatlog(internal)àfilter(qu
ickstart)ànextàgroupid (any name).**

- Then we have to add hibernate dependency in pom.xml file. And then for database connection we have to add Oracle database dependency.
- Then we have to download hibernate jar file and go to the file and choose all jar files in required file.
- Create an XML file and add

```
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"

    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
```

- Then only write < for auto syntax for hibernate properties. After providing all data in .xml file we have to go .java file
- Create an object for configuration with SessionFactory interface reference.

```
Configuration cfg=new Configuration();
    cfg.configure();

    SessionFactory factory=cfg.buildSessionFactory();
```

- Most commonly use hibernate annotations

- There are different type of mapping technique, these are

§ One to One Mapping

§ One to Many Mapping

§ Many to One Mapping

§ Many to Many Mapping

All these mapping may be Bidirectional else
Uni-directional.

Ø Fetching type:

There are two types for fetching of data, these are

§ **LAZY**

Data loading occurs when we explicitly call
getter or size method.

§ **EAGER**

Data loading occurs on the spot.

By default, the value of fetch type is always LAZY, so for fetch
the data we need to call getter or size method.

`@OneToMany(mappedBy="question",fetch=FetchType.LAZY)` and then we have
to write `System.out.println(q.getAnswers().size());`

When we use fetch type EAGER then automate answer will
execute with the execution of question.

`@OneToMany(mappedBy="question",fetch=FetchType.EAGER)`

Ø Format SQL query we have to write in hibernate.cfg.xml i.e.

```
<property name="format_sql">true</property>
```

Ø Hibernate Objects States Practical Session:

§ Transient

§ Persistent

§ Detached

§ Removed

Ø **Cascading:**

§ When we save one table data then we have to save also its related data which present in another table (mapping)

§ No need to save another table data when you use cascading keyword in hibernate.

```
@OneToMany(mappedBy="question",fetch=FetchType.EAGER,cascade = CascadeType.ALL)
```

Ø **Caching:**

§ Caching is a mechanism to enhance the performance of an application.

§ **Cache is used to reduce the number of database queries.**

§ **When we fire a query to database, if we use hibernate Caching concept then data comes from database as an object and automate store in hibernate cache memory between java application and database. Then object return to java application.**

If we again fire the same query, then same object will come hibernate cache memory if available. Here query will not fire to database.

§ **There are two types of caching,**

1. First level Caching

(Session Object, its by default provided)

2. Second Level Caching

(Session Factory, Manually Enable)

On below **first level caching** (by default) example if we fire same query two times, then only once query fire to database and object will create once.

For **second level caching** we have to use **ehcache** **dependency** and **hibernate ehcache dependency** (as hibernate core version) in pom.xml file. Then add a property attribute in hibernate.cfg.xml file. Then also add the inbuilt class from **EhCacheRegionFactory** package in property tag. Then we have to use two annotation **@Cacheable** and **@Cache**.

```
<!-- https://mvnrepository.com/artifact/net.sf.ehcache/ehcache -->
```

```
<dependency>
```

```
<groupId>net.sf.ehcache</groupId>
```

```
<artifactId>ehcache</artifactId>
```

```
<version>2.10.9.2</version>
```

```
</dependency>
```

```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-ehcache -->
```

```
<dependency>
```

```
<groupId>org.hibernate</groupId>
```

```
<artifactId>hibernate-ehcache</artifactId>
```

```
<version>5.4.5.Final</version>
```

```
</dependency>
```

```
<property name="cache.use_second_level_cache">true</property>
```

```
<property name="cache.region.factory_class">  
org.hibernate.cache.ehcache.internal.EhCacheRegionFactory  
</property>
```

```
@Cacheable
```

```
@Cache(usage = CacheConcurrencyStrategy.READ_ONLY)
```