Hibernate :

ORM 🡪 Object Relational Mapping.

ORm Tools in Market 🡪 Hibernate, IBatis, Ypplink

1. class -> create table

2. object -> represent table of rows

MVN Repo : core hiberbate for orm functionality 4.1.6

Hibernate tools in eclips :

Go to help -> eclips marketplace ->

1. jbose tools installed.

2. project- > right click -> new -> other-> hibernate-> hibernate configuration file(cfg.xml)

In main -> Configuration -> Sessionfactory -> Session -> Transaction (begin -> commit )

In hibernate.cfg.xml -> <property name=*"hbm2ddl.auto"*>update</property>

When write create then it create new table every time

In Student Pojo class -> @Entity -> allow for database and @Id -> make primary key

It show property of sql -> <property name=*"show\_sql"*>true</property>

Anotation :

@ManyToOne

@ManyToMany

@OneToMany

@OneToOne

Fetch eager rather than lazy value :

By default it is lazy make it eager by 🡪 @ManyToMany(mappedBy="student",fetch=FetchType.***EAGER***)

Hibernate Caching:

By default hibernate provide you first level cache means while maintain session we maintain this cache.

After let say we use to maintain second level of cache then we need configurations..

Types of cache : ehcache, sworm, os…

Configuration :

Pom.xml -> ecache , hibernate-ecache

Hibernate.cfg.xml -> tags

Annotation -> @cacheable -> tell tables is allow for cache or not , @cache -> used to maintain strategy of cache

Note : hibernate and hibernate-ehcache version is same..

By default - > second level of cache is only for GET data from database but Not for HQL Query.

HQL :

Hibernate Query language :

Select rollno from Student - > rollno is property and Student is class name

from Student - > entire table..

Object states persistence life cycle :

Diagram, schematic

Description automatically generated

Get v/s Load

Get is used for fetch data from database and it hits to database and create object

Load is also same use but it not hitting database if we not used it …it create proxy object.

JPA - > Java persistence API

The Java Persistence API (JPA) is a specification of Java. It is used to persist data between Java object and relational database. JPA acts as a bridge between object-oriented domain models and relational database systems.

As JPA is just a specification, it doesn't perform any operation by itself. It requires an implementation. So, ORM tools like Hibernate, TopLink and iBatis implements JPA specifications for data persistence.

We can convert our hibernate to Ibatis implementation if we use JPA.