JPA WITH HIBERNATE:

INSERT DATA:

package com.tyss.hibernateproject;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.EntityTransaction;

import javax.persistence.Persistence;

import com.tyss.JpaWithHibernate.bean.Movies;

public class InsertData {

public static void main(String[] args) {

Movies movies = new Movies();

movies.setId(10);

movies.setName("Kites");

movies.setRating(6);

movies.setReview("Drama");

EntityManagerFactory managerFactory = null;

EntityManager entityManager = null;

EntityTransaction transaction = null;

try {

managerFactory = Persistence.createEntityManagerFactory("moviesData");

entityManager = managerFactory.createEntityManager();

transaction = entityManager.getTransaction();

transaction.begin();

// to store the data EntityManagerref.persist(movies);

entityManager.persist(movies);

System.out.println("Data successfully Inserted");

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

if (managerFactory != null) {

managerFactory.close();

}

if (entityManager != null) {

entityManager.close();

}

}

}

}

-----------Start from here--------------

READ DATA FROM DB:

package com.tyss.JpaWithHibernate.bean;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

public class FindMovies {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory = null;

EntityManager entityManager = null;

try {

entityManagerFactory = Persistence.createEntityManagerFactory("moviesData");

entityManager= entityManagerFactory.createEntityManager();

// to read the values from the tables Find()

// Movies movie = entityManager.find(Movies.class, 10);

Movies movie= entityManager.getReference(Movies.class, 10);

System.out.println(movie);

//

System.out.println("Movie ID- "+ movie.getId());

System.out.println("Movie Name- "+ movie.getName());

System.out.println("Movie Rating- "+ movie.getRating());

System.out.println("Movie Review- "+ movie.getReview());

} catch (Exception e) {

e.printStackTrace();

}

finally {

if(entityManagerFactory!=null) {

entityManagerFactory.close();

}

if(entityManager!=null) {

entityManager.close();

}

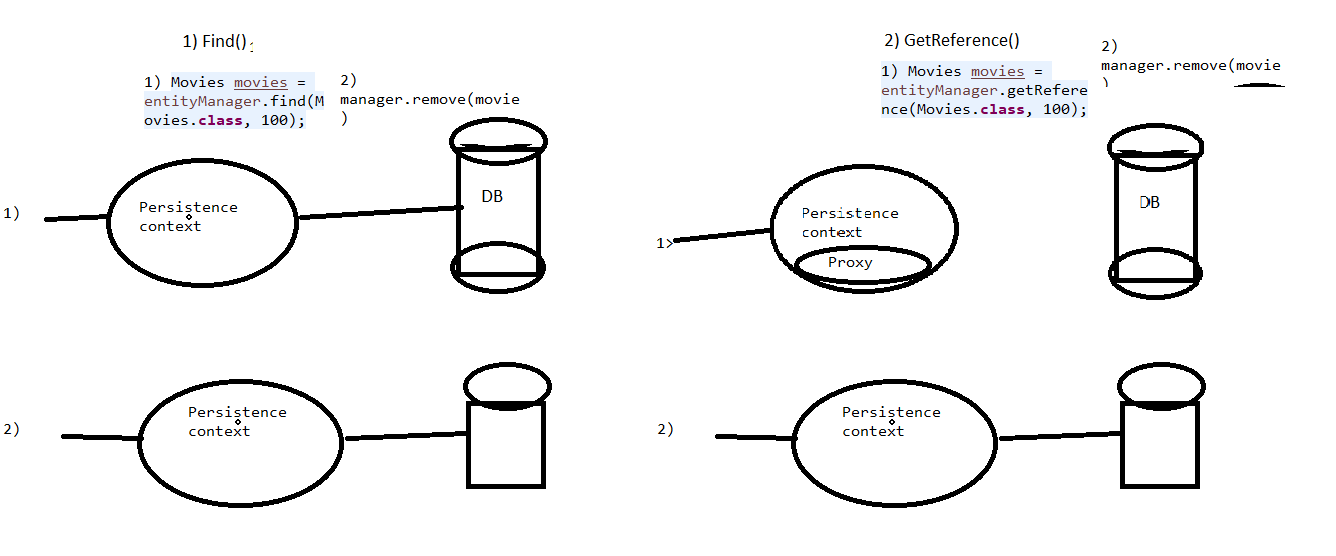
}

}

}

DIFFERENCE BETWEEN FIND() and GETREFERENCE()

Find will hit the db no matter what, getRef will hit the db only if we request for some data.



UPDATE DATA:

package com.tyss.JpaWithHibernate.bean;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.EntityTransaction;

import javax.persistence.Persistence;

public class UpdateMovies {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory = null;

EntityManager entityManager = null;

EntityTransaction entityTransaction = null;

try {

entityManagerFactory = Persistence.createEntityManagerFactory("moviesData");

entityManager = entityManagerFactory.createEntityManager();

entityTransaction = entityManager.getTransaction();

entityTransaction.begin();

Movies movies = entityManager.find(Movies.class, 10);

if (movies != null) {

movies.setName("Inception");

movies.setReview("Thriller");

movies.setRating(8.4);

}

entityTransaction.commit();

} catch (Exception e) {

if (entityTransaction != null) {

entityTransaction.rollback();

}

e.printStackTrace();

} finally {

if (entityManagerFactory != null) {

entityManagerFactory.close();

}

if (entityManager != null) {

entityManager.close();

}

}

}

}

DELETE DATA:

package com.tyss.JpaWithHibernate.bean;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.EntityTransaction;

import javax.persistence.Persistence;

public class DeleteMovies {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory= null;

EntityManager entityManager= null;

EntityTransaction entityTransaction= null;

try {

entityManagerFactory = Persistence.createEntityManagerFactory("moviesData");

entityManager= entityManagerFactory.createEntityManager();

entityTransaction= entityManager.getTransaction();

entityTransaction.begin();

Movies movies= entityManager.find(Movies.class, 10);

entityManager.remove(movies);

entityTransaction.commit();

System.out.println("Movie deleted");

} catch (Exception e) {

if(entityTransaction!=null) {

entityTransaction.rollback();

}

e.printStackTrace();

}

finally {

if(entityManagerFactory!=null) {

entityManagerFactory.close();

}

if(entityManager!=null) {

entityManager.close();

}

}

}

}

EXAMPLE OF DETACH AND MERGE:

package com.tyss.JpaWithHibernate.bean;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.EntityTransaction;

import javax.persistence.Persistence;

public class DetachMergeExample {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory= null;

EntityManager entityManager= null;

EntityTransaction entityTransaction= null;

try {

entityManagerFactory= Persistence.createEntityManagerFactory("moviesData");

entityManager= entityManagerFactory.createEntityManager();

entityTransaction= entityManager.getTransaction();

entityTransaction.begin();

Movies movie= entityManager.find(Movies.class, 10);

System.out.println(entityManager.contains(movie)); //true

entityManager.detach(movie);

System.out.println("After detach");

System.out.println(entityManager.contains(movie)); //false

Movies mergedData= entityManager.merge(movie);

mergedData.setName("Kites");

System.out.println("After merge");

System.out.println(entityManager.contains(mergedData)); //true

// entityManager.remove(movie); //show them that after detach remove doesn't work

// entityManager.remove(mergedData); //show them that after merge remove works

entityTransaction.commit();

} catch (Exception e) {

if(entityTransaction!=null) {

entityTransaction.rollback();

}

e.printStackTrace();

}

}

}