**package** com.pms.model;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.NamedQueries;

**import** javax.persistence.NamedQuery;

**import** javax.persistence.Table;

@Entity

@Table

//@Cachable

//@Cache(strategy="") //readonly

@NamedQueries({

@NamedQuery(name = "viewAllAccounts", query = "FROM Account a") ,

})

**public** **class** Account {

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

Integer id;

@Column

String name;

@Column

Double amount;

@Override

**public** String toString() {

**return** "Account [id=" + id + ", name=" + name + ", amount=" + amount + "]";

}

**public** Account(String name, Double amount) {

// super();

// this.id = id;

**this**.name = name;

**this**.amount = amount;

}

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Double getAmount() {

**return** amount;

}

**public** **void** setAmount(Double amount) {

**this**.amount = amount;

}

**public** Account() {}

}

====Client

**package** com.pms.client;

**import** java.util.List;

**import** javax.persistence.TypedQuery;

**import** javax.persistence.criteria.CriteriaBuilder;

**import** org.hibernate.Criteria;

**import** org.hibernate.Query;

**import** org.hibernate.SQLQuery;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.criterion.Projections;

**import** org.hibernate.criterion.Restrictions;

**import** org.hibernate.service.ServiceRegistry;

**import** com.pms.model.Account;

**public** **class** HibernateAccountClient {

**public** **static** **void** main(String[] args) {

// Configuration configObj = new Configuration(); //1

// configObj.configure("hibernate.cfg.xml");

// Since Hibernate Version 4.x, ServiceRegistry Is Being Used //2

// ServiceRegistry serviceRegistryObj = new StandardServiceRegistryBuilder().applySettings(configObj.getProperties()).build();

// Creating Hibernate SessionFactory Instance //3

SessionFactory sessionFactory =**new** Configuration().configure("hibernate.cfg.xml").~~buildSessionFactory~~();

Session session = sessionFactory.openSession(); //4

Transaction tx=session.getTransaction();

tx.begin();

// Account a1=new Account("Ram123",10000.00); //transcient

//insert

// session.save(a1); //insert into account values..... //persistent

// Account a2=new Account("Ram1234",10000.00); //transcient

// session.save(a2);

//update

/\* Account a3=(Account)session.get(Account.class, 3);

a3.setName("Ram3333");

session.update(a3);

session.flush();

\*/

//delete

//Account a3=(Account)session.get(Account.class,2);

//Account a3=(Account)session.load(Account.class,5);

// session.delete(a3);

//Query q=session.createQuery("from Account a"); //HQL

// Query q=session.createQuery("from Account a where a.id='3'"); //HQL

//SQLQuery q=session.createSQLQuery("select \* from account");//native SQL

//q.addEntity(Account.class);

// List<Account> alist=q.list();

//:named parameters

//Query query = session.createQuery("UPDATE Account SET amount = ? WHERE id = ?");

//query.setParameter(2, 4);

//query.setParameter(1, 39999.99);

/\* Query query = session.createQuery("UPDATE Account SET amount = :amt WHERE id = :id");

query.setParameter("id", 4);

query.setParameter("amt", 39999.99);

query.executeUpdate();

\*/

//alist.stream().forEach(a->System.out.println(a.getId()+" "+a.getName()));

Criteria cri = session.createCriteria(Account.**class**);

//Criteria cr = session.createCriteria(Employee.class);

//cri.add(Restrictions.eq("name", "Ram123"));

// cri.add(Restrictions.gt("amount", 5000.00));

cri.setProjection(Projections.*property*("amount"));

List<Double> list1=cri.list();

// Query q=session.createQuery("from Account a"); //HQL

// List<Account> alist=q.list();

//viewAllAccounts

Query q=session.getNamedQuery("viewAllAccounts");//Named query

List<Account> list2=q.list();//data is retrived into appcation heap from db

//session level cache / first level cache

**for**(Account a1: list2) {

System.***out***.println(a1.getId() +" "+a1.getName()+" "+a1.getAmount());

// System.out.println(a1);

}

session.close();

Session session1 = sessionFactory.openSession(); //4

**for**(Account a1: list2) {

System.***out***.println(a1.getId() );

// System.out.println(a1);

}

**for**(Account a1: list2) {

System.***out***.println(a1.getId() +" "+a1.getAmount());

// System.out.println(a1);

}

// TypedQuery<Account> typedQuery = session.createSQLQuery("select \* from account");

//5th row is the first row in the result set

// typedQuery.setFirstResult(2);

//Starting from the 5th record, and retrieve the next 10 records

// typedQuery.setMaxResults(3);

// List<Account> next3 = typedQuery.getResultList();

/\* for(Account a1: next3) {

System.out.println(a1.getId() +" "+a1.getName()+" "+a1.getAmount());

// System.out.println(a1);

}

\*/

//CriteriaBuilder cb = session.getCriteriaBuilder();

//CriteriaQuery<Item> cr = cb.createQuery(Item.class);

//cri.setProjection(Projection.);

// The class passed into createQuery() indicates that the results of this CriteriaQuery will be Student objects

**try** {

}

**catch** (Exception e) {

tx.rollback();

}

**finally** {tx.commit();

}

session1.close();

// Account a1=new Account("Ram123",10000.00);

//a1.setName("cccc");

// detached

}

}