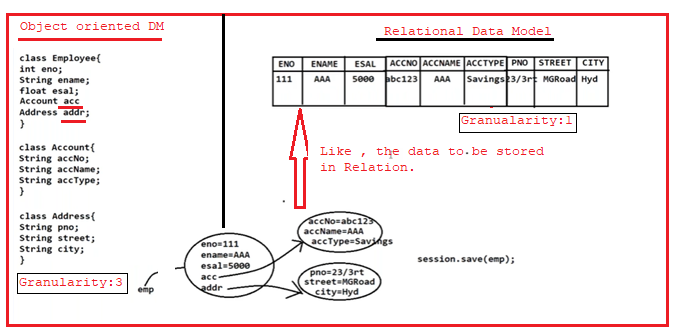
**Granualarity-mismatch :-** From Below context, the no.of classes in object oriented Data model and no.of tables in relational Data model is not matched.

In object oriented Data model, one object property represents ‘n’ number of columns in relation in relational Data model.

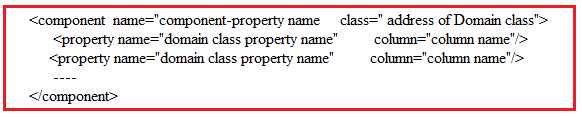
The above two cases leads to granualarity-mismatch.

It reduces the Data persistency. To improve the Data persistency , we should use “Hybernate”. The hibernate provided solution to granualaity-mismatch in the form of “**component –mapping” .**



Where **acc, addr** are component properties.

**1.Mapping between component property and columns in Relation using xml tags:**



This <component> tag must be written in <class>tag.That means <component> tag is child tag to <class> tag.

**2.Mapping between component property and columns in Relation using Annatations:**  The following tags are used

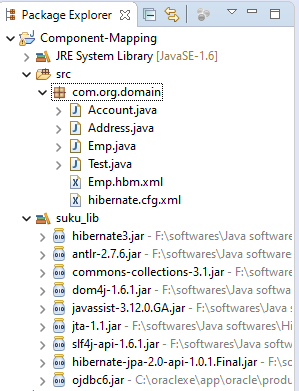
[1.@Embedded](mailto:1.@Embedded)

[2.@Embeddable](mailto:2.@Embeddable)

2.1. @Embedded:- The @Embeddedannotation is used to mark a field in a domain class as being an embeddable object.

2.2 @Embedded:- The @Embeddable annotation is used to mark a class as being embeddable, meaning its properties can be included in another class as a value type. The class marked with @Embeddable is called the embeddable class.

Example:1 Component mapping using XML tags.



**Mapping File:- Emp.hbm.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping>

<class name=*"com.org.domain.Emp"* table=*"Emp"*>

<id name=*'eno'* column=*'eno'*/>

<property name=*'ename'* column=*'ename'*/>

<property name=*'esal'* column=*'esal'*/>

**<component name=*'acc'* class=*"com.org.domain.Account"*>**

**<property name=*'accNo'* column=*'accNo'*/>**

**<property name=*'accName'* column=*'accName'*/>**

**<property name=*'accType'* column=*'accType'*/>**

**</component>**

**<component name=*'addr'* class=*"com.org.domain.Address"*>**

**<property name=*'pno'* column=*'pno'*/>**

**<property name=*'street'* column=*'street'*/>**

**<property name=*'city'* column=*'city'*/>**

**</component>**

</class>

</hibernate-mapping>

**Hibernate Configuration file:hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

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

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<mapping resource=*"com/org/domain/Emp.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

**Domain classes :**

**1.Emp.java**

**package** com.org.domain;

**public** **class** Emp {

**private** **int** eno;

**private** String ename;

**private** **float** esal;

**private** Account acc;

**private** Address addr;

**public** **int** getEno() {

**return** eno;

}

**public** **void** setEno(**int** eno) {

**this**.eno = eno;

}

**public** String getEname() {

**return** ename;

}

**public** **void** setEname(String ename) {

**this**.ename = ename;

}

**public** **float** getEsal() {

**return** esal;

}

**public** **void** setEsal(**float** esal) {

**this**.esal = esal;

}

**public** Account getAcc() {

**return** acc;

}

**public** **void** setAcc(Account acc) {

**this**.acc = acc;

}

**public** Address getAddr() {

**return** addr;

}

**public** **void** setAddr(Address addr) {

**this**.addr = addr;

}

}

**2. Account.java**

**package** com.org.domain;

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

**private** **int** accNo;

**private** String accName;

**private** String accType;

**public** **int** getAccNo() {

**return** accNo;

}

**public** **void** setAccNo(**int** accNo) {

**this**.accNo = accNo;

}

**public** String getAccName() {

**return** accName;

}

**public** **void** setAccName(String accName) {

**this**.accName = accName;

}

**public** String getAccType() {

**return** accType;

}

**public** **void** setAccType(String accType) {

**this**.accType = accType;

}

}

**3.Address.java**

**package** com.org.domain;

**public** **class** Address {

**private** **int** pno;

**private** String street;

**private** String city;

**public** **int** getPno() {

**return** pno;

}

**public** **void** setPno(**int** pno) {

**this**.pno = pno;

}

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

}

**Client Application:Test.java**

**package** com.org.domain;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**public** **class** Test {

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

Configuration cfg=**new** Configuration();

cfg.configure("com/org/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Address addr=**new** Address();

Account acc=**new** Account();

Emp e=**new** Emp();

Transaction t=s.beginTransaction();

addr.setCity("nlr");

addr.setPno(77);

addr.setStreet("Rajaka street");

acc.setAccName("suku");

acc.setAccNo(77777);

acc.setAccType("saving");

e.setEno(1);

e.setEname("suku");

e.setEsal(7000);

e.setAcc(acc);

e.setAddr(addr);

**int** pk=(Integer)s.save(e);

t.commit();

**if**(pk==1) {

System.***out***.println("Record Inserted");

}

**else**

{

System.***out***.println("Record not inserted");

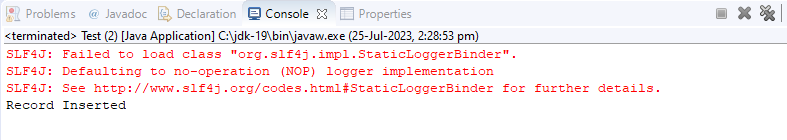
}

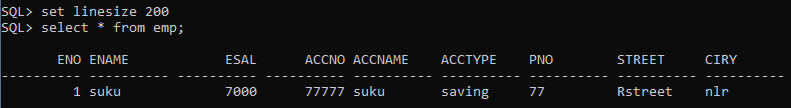
}

}

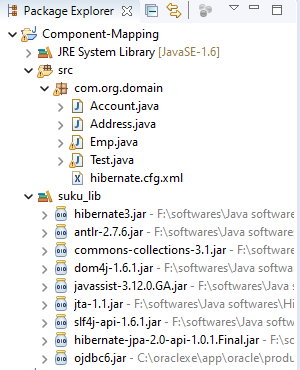
Output:-







Example:2 Example:1 Component mapping using Annotated class.



**Domain classes and Annotated classes:**

**1.Emp.java**

**package** com.org.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Embedded;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="emp")

**public** **class** Emp **implements** Serializable {

@Id

@Column(name="eno")

**private** Integer eno;

@Column(name="ename")

**private** String ename;

@Column(name="esal")

**private** **float** esal;

@Embedded

**private** Account acc;

@Embedded

**private** Address addr;

**public** **int** getEno() {

**return** eno;

}

**public** **void** setEno(**int** eno) {

**this**.eno = eno;

}

**public** String getEname() {

**return** ename;

}

**public** **void** setEname(String ename) {

**this**.ename = ename;

}

**public** **float** getEsal() {

**return** esal;

}

**public** **void** setEsal(**float** esal) {

**this**.esal = esal;

}

**public** Account getAcc() {

**return** acc;

}

**public** **void** setAcc(Account acc) {

**this**.acc = acc;

}

**public** Address getAddr() {

**return** addr;

}

**public** **void** setAddr(Address addr) {

**this**.addr = addr;

}

}

**2.Account.java**

**package** com.org.domain;

**import** javax.persistence.Column;

**import** javax.persistence.Embeddable;

**import** javax.persistence.Entity;

@Entity

@Embeddable

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

@Column(name="accNo")

**private** **int** accNo;

@Column(name="accName")

**private** String accName;

@Column(name="accType")

**private** String accType;

**public** **int** getAccNo() {

**return** accNo;

}

**public** **void** setAccNo(**int** accNo) {

**this**.accNo = accNo;

}

**public** String getAccName() {

**return** accName;

}

**public** **void** setAccName(String accName) {

**this**.accName = accName;

}

**public** String getAccType() {

**return** accType;

}

**public** **void** setAccType(String accType) {

**this**.accType = accType;

}

}

**Address.java**

**package** com.org.domain;

**import** javax.persistence.Column;

**import** javax.persistence.Embeddable;

**import** javax.persistence.Entity;

@Entity

@Embeddable

**public** **class** Address {

@Column(name="pno")

**private** **int** pno;

@Column(name="street")

**private** String street;

@Column(name="ciry")

**private** String city;

**public** **int** getPno() {

**return** pno;

}

**public** **void** setPno(**int** pno) {

**this**.pno = pno;

}

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

}

**HibernateMapping file: hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

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

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<mapping class=*"com.org.domain.Emp"*/>

</session-factory>

</hibernate-configuration>

**Client Application: Test.java**

**package** com.org.domain;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

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

**public** **class** Test {

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

Configuration cfg=**new** Configuration();

cfg.configure("com/org/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Address addr=**new** Address();

Account acc=**new** Account();

Emp e=**new** Emp();

e=(Emp)s.get(Emp.**class**, 1);

System.***out***.println(" =========Employee Details ==========");

System.***out***.println("eno:"+e.getEno());

System.***out***.println("ename:"+e.getEname());

System.***out***.println("esal:"+e.getEsal());

System.***out***.println("===========Account Details===========");

System.***out***.println("Accno"+e.getAcc().getAccNo());

System.***out***.println("AccName"+e.getAcc().getAccName());

System.***out***.println("AccType"+e.getAcc().getAccType());

System.***out***.println("===========Address Details============");

System.***out***.println("pno"+e.getAddr().getPno());

System.***out***.println("street"+e.getAddr().getStreet());

System.***out***.println("city"+e.getAddr().getCity());

}

}

**Output:**-

