Module 1 : core java application

accepts user name and training name from **the user** and will use **TraineeBean** to insert the new Trainee detail into Database. User id for the user should be generated automatically through sequence.

accepts user name and training name from **the user through Scanner**

**input validation : done thru reg ex**

**[A-Z][a-z]{19}**

**Validation was done in service layer**

**But in core java test, u inserted into hashmap and not the table**

Web based application

accepts user name and training name from **the user thru html form**

**input validation : will be required . u will learn how to do it in the html form**

in module 2; u will insert into table using JPA and not JDBC

in java code, we come across the following two concepts

1. is a relation : inheritance ; child class; parent class

Class Employee{

//generic attributes applicable to all employees

}

Class Manager extends Employee{

//will have all the attributes of employee

//may have other attribtues specific only to managers

}

When u work with jpa in such a scenario, u will wonder how many tables to create in the database side

1. should u create only one table containing columns from both the tables

Single table gets created with the name u specify, it contains columns from both the class plus one extra column whose name u can determine; this is the discriminator column as hibernate would want to maintain info about the class whose object is being persisted

What value should hibernate store in this column is called the discriminator type

when object of class employee is persisted and when object of class manager is persisted ?

for your example u have chosen that discriminator type should be string

2. should u create two independent tables; one for each class; one table per class

3. should u create two tables ; but which are joined by a common column

Jpa gives u three strategies ; one for each of the points 1,2,3

The third one is what u observe in oracle and what v r familiar with

Database concept : if two tables are related, how to determine which is the parent and which is the child

Emp and dept table

Emp table

Deptno

empno

Dept table

2. has a relation : when object of one class is contained within object of another class

Class A{

B b; // this can be one object or a collection of objects

}

//following is a one to many relation

Class A{

List<B> b;

}

One to one : one employee belongs to one department at a time

Unidirectional : only one class has reference of other class

Class Student

{

Address address;

}

Class Address{

//no info about student ; hence this is called one to one relationship unidirectional

}l

Execute the one to one unidirectional and tel me how many tables get crated

Two new annotations are required here

Cascade concept is of database not of JPA

Cascade means : if two tables are related then inserting into one table will automatically insert into other table

U inserted record into student table; record automatically got inserted into address table also

Similarly for updating/deleting

Observation on executing one to one unidirectional project

1. how many tables got created : two

2. how many rows got inserted : one in each table for one execution

2 imp annotations

@onetoone

@joincolumn

Bidirectional

Both classes have references of each other ; same eg of student and address

{

}

//to create bi-directional relationship, use one to one with mappedBy

//mappedBy attribute indicates property name of owner i.e. Student class

@OneToOne(mappedBy="address")

Observation on executing one to one BIdirectional project

1. how many tables got created :

2. how many rows got inserted :

OBSERVATION

The client program of the uni and bi directional one to one looks similar

So what advantage does bi directional have over unidirectional ?

Ans : in unidirectional : if u fetch the student record from the table; u can get his /her corresponding address by the below code

Student stu = (Student ) Entitymager.find(Student.class,1000);

Syso(stu.getAddress)// with this line u can fetch student ka address

But if u fetch an address , u will not be able to find out which student it belongs to

Address addr =(Address)Entitymager.find(Address.class,1234);

Syso(addr.getStduent() will not be possible as address does not have a student object

Hope u have understood

The above is solved by bidirectional relationship

i.e if u have a student record, u can print his address. Even if u have an address object; u can print which student it belongs to

end of story . thank u 😊

One to many : one department has many employees

Class Dept{

List<Employee> emps;

}

Many to many : one product has many ingredients. One ingredient is used in many products

When both tables give u an answer of one to many; it results in many to many relationship

One order can have many products; o

One product can be present in many orders

A third table called “mapping/join table” gets created in many to many relationship

This new table has 2 columns ; both of them would be the primary key of both the individual tables

FetchType

Lazy fetch : select the records from the table only if its is asked for . eg student.getAddress()

Em.find(Student.class,100) : this method will only retrieve student record with code 100

syso(student) ;//will not have the address details

If the student has address associated with it, it wont be retrieved

student.getAddress() has to be written to fetch the address details

Get the address details of the student only when this method is written

Eager fetch : Em.find(Student.class,100) : this method will retrieve student record with code 100 and also address records of this student. No need to explicitly write student.getAddress()

Syso(student) ;//will print address info also provided tostring is present in the student class

Class Order{

Products [] products; // one order has many products

}

Class Employee{

}

Oracle concepts

Determine the common column between the two tables emp and dept

Ans : deptno

Ask yourself this questions . in which table common column values can repeat

Ans :u get ans as no for dept table. then this table becomes the parent table

Ask the same question to emp table

Can deptno column values repeat in the emp table :

Ans: yes. Then emp table is the child table between emp and dept

Many employees working in one dept

Each table needs a primary key

Which is the primary key in dept table ; deptno

And the primary key in emp table is : emp no

What are the available deptnos in the dept table ? ans 10,20,30,40

Can u enter value 100 for deptno column in emp table ? no

How to ensure that value entered for deptno column in emp table is a valid deptno ? ans : foreign key

Insert into emp(empno,ename,deptno) values(8989,’kavita’,100);

If u apply a foreign key constraint; oracle will ensure that value entered for deptno column in the emp table must be present in the deptno column of dept table