

Hibernate

Hibernate is a one of the implementation of ORM tool provided by Jboss.

Database table : relation in database.

Employee

Id(PK) name salary

Now we need to create java bean class that class map to table. In orm that java bean class is known as **entity** class.

Old version in hibernate we were/are using xml file to do the mapping

Table must be map to java bean class

All variable must be map to all columns with their data types.

The column which contains primary key we need to mention in xml file.

@Entity annotation on java bean class. this annotation is use to map table using ORM.

@Table annotation: this annotation is use to map to table if java bean or entity class name and table may be different.

@Id this annotation we need to on that property. That property must be map to primary key columns.

@Column : this annotation is use to map with variable and column name if both have different names.

In hibernate we write database connection details in `hibernate.cfg.xml` ie configuration.

This file hold the database details like driver name, url, username, password and mapping class ie entity class details.

Hibernate provided pre defined class ie

Configuration which help to load the hibernate.cfg.xml file which contains Database details, dilates class and mapping file information.

SessionFactory : it is an interface provided by hibernate which provide set of method which help to create session object. SessionFactory is equal to Connection in jdbc.

Session : Session is an interface which provide set of method which help to do some operation on entity class like save, delete, update and find etc.

Session is like a Statement or PreparedStatement in jdbc.

Session object we can create with help of SessionFactory reference.

If we do any DML operation using jdbc by default they are auto commit.

In jdbc if we want to transaction concept then we need to disable auto commit option using `con.setAutoCommit(false);`

Then after executeupdate("DML Query ")

You can use `con.commit()` or `con.rollback();`

If we do any DML operation using ORM tool like hibernate or jpa it is not by default auto commit.

If we want to use transaction in hibernate. Hibernate provided pre defined api it Transaction.

To retrieve more than one record hibernate provided their own query language ie HQL (Hibernate Query Language)

SQL : Database dependent query language

It retrieve the record in string or query format.

Select * from employee; employee is table name and it is not case sensitive. * means all column from a table.

Select * from employee where id = 101;

Select * from employee where salary > 34000;

Id and salary are column name.

Select name from employee

Select id from employee

Select id, name from employee

HQL : Database independent query language

It retrieve object in programming language side.

Select emp from Employee emp; here Employee is entity/java bean class name case sensitive and emp is object. emp is object which help to retrieve all variables values.

Select emp from Employee emp where emp.id = 101;

Select emp from Employee emp where emp.salary > 34000;

emp.id, emp is object and id is entity variable name.

retrieve partial object

select emp.id from Employee emp; retrieve only id variable

select emp.name from Employee emp; retrieve only name variable

select emp.id, emp.name from Employee emp; retrieve id and name variable

Spring MVC with ORM ie Hibernate

Create dynamic web project with version 2.5 with tomcat and convert this project to maven project.

Spring mvc @Controller with @service , @repository, @autowried

Hibernate : hibernate API ie sessionFactory, session, transaction etc

Spring jdbc : data source

My sql connector : to connect the database.

Spring orm : it help to integrate with orm tool like hibernate or jpa.

LocalSessionFactoryBean: it is a pre defined class part of spring orm which help to integrate with hibernate orm tools. This class help use to do DI for sessionFactory in dao class.

Hibernate provide a features which help to create the table if table not present.