

1. What are the types of inheritance models and describe how they work like vertical inheritance and horizontal ?

Ans. (a)

There are three types of inheritance mapping in hibernate

1. Table per concrete class with unions
2. Table per class hierarchy
3. Table per subclass

Example:

Let us take the simple example of 3 java classes.

Class Manager and Worker are inherited from Employee Abstract class.

1. Table per concrete class with unions

In this case there will be 2 tables

Tables: Manager, Worker [all common attributes will be duplicated]

2. Table per class hierarchy

Single Table can be mapped to a class hierarchy

There will be only one table in database called 'Employee' that will represent all the attributes required for all 3 classes.

But it needs some discriminating column to differentiate between Manager and worker;

3. Table per subclass

In this case there will be 3 tables represent Employee, Manager and Worker

Ans. (b). Table per concrete class using implicit polymorphism can also be added as inheritance mapping in hibernate

2. what is lazy fetching in hibernate ?

Ans 1

Lazy setting decides whether to load child objects while loading the Parent Object. You need to do this setting respective hibernate mapping file of the parent class. Lazy = true (means not to load child) By default the lazy loading of the child objects is true. This make sure that the child objects are not loaded unless they are explicitly invoked in the application by calling getChild() method on parent. In this case hibernate issues a fresh database call to load the child when getChild() is actually called on the Parent object. But in some cases you do need to load the child objects when parent is loaded. Just make the lazy=false and hibernate will load the child when parent is loaded from the database. Examples lazy=true (default) Address child of User class can be made lazy if it is not required frequently. lazy=false But you may need to load the Author object for Book parent whenever you deal with the book for online bookshop.

Ans 2

Lazy fetching is related to loading of the Child objects for its parent (In the terms of database its primary key-foreign key relationship). In hbm.xml file you have to specify whether you want to load the child objects while loading the parent. By default hibernate doesn't load the whole child objects (lazy="true").

But Sometimes it doesn't work properly, it doesn't load the child objects for the parent.and second problem is that if you have condition like

Employee Table -1-----n-> Emp_Dept Table<-n-----1- Department Table

And you have many to many relationship between them, then by specifying lazy="false" for both the parent object(Employee And Department) hibernate try to get all the child for this parent and for this child. Now this child becomes the parent and hibernate will try to get all the child for this parent. This process may continue.This results in performance issue.

So U have to take care while writing the mapping files in case of many-to-many relationships.

Ans 3

Simply saying: It maintains the relationship between two tables.

Ans 4.

There are two types of Loading in application. Eager loading and Lazy loading. In eager loading we will fetch all the values from the Persistent storage and cache it. It will make serious performance issues. There we use lazy loading to avoid that scenario.

Some times we don't need to load the child table values, In that case we have to us lazy = true in .hbm file. so hibernate will fetch only parent table values. Internally it will load the wrapper class but it does not cache all the values till we perform operation.

Main

Advantage: It will avoid caching unnecessary values and improve performances.

3. How JPA and hibernate are related as per EJB 3 specifications?

Ans 1. JPA is a persistence API by Sun. As per sun, JPA takes best ideas from persistence frameworks like Hibernate, TopLink2.

Ans 2. JPA Is Java Persistance API , it will give the most pefect api from the Persistanc Frame work like, Top-link, Hibernate , OracleDAO. This was not there with ejb2.1 . The main feature of EJB3.0 is the favour of JPA so that programers can use the entity bean much easier and simpler.

Ans. 3. Java Persistence API is implemented as per EJB3 specification and having more features of Hibernate. JPA is framed by adding few features of kodo which is a ORM tool.

Ans. 4. JPA is official acceptance from SUN about its failure on EJB. It has to abandon its EJB model to go to ORM model. Finally, Sun should provide developers some tools to migrate Java programs to .net easily with out pain

4. What is component mapping in hibernate?

Ans. 1.

A component is a contained object that is persisted as a value type ,not an entity reference.eg)

```
public class person{
```

```
    private Name name;
```

```
    public Name getName(){ return name;}
```

```
    public void setName(Name name){this.name=name;}
```

```
.....
```

```
}
```

```
public class Name
```

```
{char initial;
```

```
    String first;
```

```
    String last;
```

```
    public char getInitial(){return initial;}
```

```
    public void setInitial(char initial){this.initial=initial;}
```

```
.....//first,last
```

```
}
```

Now 'Name' may be persisted to the component of 'person'

in hbm:

```
<class name="eg.Person" table="person"> <id name="Key" column="pid" type="string">
<generator class="uuid"/> </id> <property name="birthday" type="date"/> <component
name="Name" class="eg.Name"> <!-- class attribute optional --> <property
name="initial"/> <property name="first"/> <property name="last"/> </component></class>
```

The person table would have the columns pid, birthday, initial, first and last.

5. How do you handle the situation where persistence class attribute name is one of the reserved keyword in database; e.g. user ? Will this cause any issue while hibernate executes SQL statements behind the scene?

Ans. 1.

you should not declare java keywords as attribute name .because when u execute the code it will search for getter /setter method.which can not be created in java files(reserved keywords);

Ans. 2.

It will not create any problem hibernate will take care of it. In table columns if table can be created with that attribute name then it can be use in hibernate also.

The question is not java reserved key word. Pls read the questions carefully b'fore replying

Ans. 3.

It doesnt make any problem, Through hsql u can perform the query. If it is resrvd key word also, hibernate will take care of it.

Ans. 4.

In case attribute name in a class is keyword in underlying database, attribute should be mapped with some other column name that is acceptable for DB. For instatce a class can have attribute "date" (getDate()/ setDate()) and in most of DB, date is reserve keyword for "Date" datatype. So we can map this with some other column name

```
<property name="date" column="blah_date" type="date" />
```

Otherwise mapping won't work.

6. Why do you need ORM tools like hibernate?

Ans. 1.

To overcome the "paradigm mismatch" between object oriented data and table oriented relational databases.

Ans. 2.

Lot of lot advatages are giving by hibernate like tools. It will also give an clear OObview . It will help us to

create,fetch,Update and delete using simple pOJo(plain java Object).

Ans. 3.

The hibernate ORM's main benefit comes when we are using collection of databases or when we want to move from one database to the other, all we gotta do is get into hibernate property file change the settings that it, its ready to go.

Ans. 4.

ORM Tools provide a slick way of persisting objects (data) to a database. ORM tools greatly reduce the need for the developers having to write SQL. ORM tools honor complex relationships. Thus if you have an object that has child objects on it (i.e. a customer who can have many addresses) when you issue a save on the parent, the ORM tool is smart enough to save all objects and will ensure they are wrapped in a transaction.

Ans. 5.

Basically we need Hibernate like tools for the purpose of developing the application in short period of time. The product like Hibernate used for the productivity. When large amount of data have to handle that time instead of using JDBC we used Hibernate.

7. What is the difference between sorted and ordered collection in hibernate?

Ans. 1.

A sorted collection is sorted in-memory using java comparator, while ordered collection is ordered at the database level using order by clause.

8. what is the main advantage of using the hibernate than using the sql

Ans. 1.

main advantage is that it avoids writing queries.ofcourse, u have to write to some extent but a lot is relaxed.most of the work is taken care by mapping.also criteria class is very useful for complex joins.

Ans. 2.

The main advantage is ..using ORM we can avoid the jdbc API completely and also provides ease to the developer in developing the classes.

Ans. 3.

It will give more performance than SQL Written Query. But it doesn't mean that it will replace by hibernate.

When and how the hibernate tool is selecting is an important role. One who is not strong in SQL Select hibernate to avoid SQL is not good practice. First of all we need basic knowledge of SQL and its working.

Otherwise he would fail to understand the process happening on the underneath , it will cause performance issues and make rework also.

Ans. 4.

Easily migrate your code between different databases. Good for updating, maintaining your application.

Ans. 5.

The main advantage of using hibernate is there reduces writing huge code .

Actually hibernate developers developed for easy use not to write huge code.

Here there is no need to write the jdbc code here we will use the connection pooling technique and it happens internally and because of connection pooling we can reuse the connection from the pool.

And the relationship will be between names of the properties and the names of the columns and the names of classes to names of tables.

Ans. 6.

I think there is one big advantage of hibernate, that is you can make Database independent application. As we have to work on POJO class for interacting with the database, so its basically reduce dependency on JDBC.

Ans. 7.

Hibernate is based on object oriented concept like java. so it has better compatibility with java than sql. In Jdbc we have to manually handle exception and every time we need to open/close the connection,create/close the statement , resultset whatever we have used.

These things are taken care by hibernate.We need not bother about this.

Hibernate uses ORM which is a better approach.

In Jdbc we use .properties file and in hibernate we use .xml file ,which is a better approach.

In Hibernate we can use collection,mapping which are the advantages.

Some advanced features of Hibernate are :

Object-oriented query language

Transparent persistence without byte code processing

Automatic primary key generation

Object/Relational mapping definition

High performance

9. how to create primary key using hibernate?

Ans. 1.

In your hbm.xml file use field.this field create a primary key for u .

Ans. 2.

id field in hbm.xml file is used to specify the primary key in database. We also use generator to specify the way primary key is generated to the database. For example

```
< id name="testId" type="string" >
< column name="testColumn" length="40" / >
< generator class="increment" / >
< /id >
```

here the primary key field name in database is testColumn and it will automatically incremented by one as

the generator is specified as increment.

Ans. 3.

In .hbm file there is a tag called id, there we mention the primary key.

```
<hibernate-mapping>
<class name="pojo1" table="pojo1" discriminator-value="s">
<id name="empid" type="string" >
<generator class="assigned"/>
</hibernate-mapping>
```

Here Id ="empid", that will act as primary key of the table "pojo1"

10.what is the advantage of Hibernate over jdbc?

Ans. 1.

As such you cannot compare Hibernate with JDBC.

One uses JDBC calls internally in the Hibernate files to get connections for the Databases.

Ans. 2.

Hibernate is used to persist the objects in the database, but we have to use the jdbc to connect to database.
JDBC is used to store the primitives in the database.

Ans. 3.

Hibernate is basically a ORM tool which allows you to perform database activies without bothering about the Database change.

You dont need to change the SQL scripts if you change database.

Apart from that you dont need to write most of the SQL scripts for persisting ,deleting object and parsing the resultsets.With respect to perfomance, hibernate provide the capability to reduce the number of database trips

by creating the batch processing and session cache and second level cache.

It also supports the transactions.

More than this all, it is very easy to make a cleaner separation of Data Access Layer from business logic layer.

With all the capabilities mentioned above it is fast and easy to learn hibernate, develop application and maintain easily.

Ans. 4.

The core drawback of JDBC is that it doesn't allow you to store objects directly to the database you must convert the objects to a relational format. For instance, if you want to persist a new instance of the Emp class to the database, you must first convert the Emp object to a SQL statement that can be executed on the underlying database. Similarly, when rows are returned from the database, you must convert each result row into an instance of Event.

Ans. 5.

It is a ORM tool used for OR Mapping which is not possible using JDBC.

Ans. 6.

1. Hibernate is not tightly tied with any underlying database. Whereas JDBC is tightly tied with the underlying database.

"Write Once persist anywhere" using hibernate can be achieved by changing the dialect in configuration

xml file.where as in JDBC we need to change the code.

2)The automatic mapping of Java objects with database tables and vice versa is called Transparent Persistence. Hibernate provides transparent persistence and developer does not need to write code explicitly to map database tables tuples to application objects during interaction with RDBMS. With JDBC this conversion is to be taken care of by the developer manually with lines of code.

11.Q1: What is the difference between Hibernate and EJB 2.1?

Q2: Which one, among EJB 2.1 and Hibernate, is better?Explain with reason.

Q3: When to use EJB 2.1 and when to use Hibernate? Explain with scenarios.

Ans1: hibernate is a ORM(object relation mapping) tool which can be used for creating a mapping between plain java bean objects (POJO) and a persistent storage (rdbms).The EJB 3.0 specification is divided into two parts the first which deals with session and MDBs and the second which deals with persistence and entity beans. The latter part is called JPA(java persistiance API). HIbernate 3.0 implements the JPA specification.EJB 2.1 is a specification for defining loosely coupled reusable business components.ans 2 & 3) EJB 2.1 and hibernate serve two different purposes. Hibernate can be co related with the entity beans in EJB 2.1.HIbernate offers far more extensive features then plain entity beans.still there are no containers (applicaiton servers) available which fully implement the EJB 3.0 specification. depending upon the buisness needs hibernate framework can be used in conjuction with EJB2.1 to achieve the JPA abstraction.

Ans. 2.

Basically Ejb and Hibenate is enterly different one But having realtion with Entitybean in Ejb and Hibenate.

Entity bean is also used for object orientd view . It need lot things to configures fo make it possible and Lot of coding need for it. After all the performance is also very few. In ejb eager loading approch has taken for loading it will cause serious performance issues. In java the most concept is polymorphisum and inheritance. We cant get the favour of these concept .

For avoiding it new approch had taken thats called hibernate. Using pojo(plain java object) and .hbm file we can make it possible. There is having all the advantage of Object oriented view.

12.what is the difference between beans and hibernate(which one is good)?

Ans. 1.

Beans are at form level and hibernate at data base level.

Ans. 2.

Hibernate adapts beansIts plain old Java objects. POJOHibernate als has a bean orientation.But the Bean is just not persistent, usng hibernate you can make a bean persistent.A bean when used in Hibernate It can create a table ,Add a row,Delete a rowUpdate row.THis is all done with Hib Configuration stuff.Hibernate is ORM as it can relate a bean to a row in a table

Ans. 3.

Its is one of the most popular ORM technology available to date in the market.

Hibernate uses POJO classes for handling persistence, these classes need to adhere normal bean writing (not enterprise beans) rules.

The same mapping can be used for different database server by just switching the delegate and driver details in hibernate.cfg.xml file.

This comes with its own framework , we need not require an application server to deploy and test it . We can run as a normal java program .It has some built in utilities which reduces application development time .

Java programmers now need not worry about SQL now .

It even comes with its own transaction management utility.

If we go for beans i.e., enterprise beans . Now java programmers need to handle the persistence managment using JDBC . SQL is not an easy for all the programmers. Atleast programmers need to integrate the queries written by the DBA's . Enterprise beans provides CMP but the overhead here is we need to have the Application Server for its deployment and we are forced to use the EntityBeans for this reason an d which might be an overhead in some of the applications .

We have more differences and advantages of hibernate over beans for further reference please reffer to Hibernate site

Ans. 4.

It's one of the most popular ORM technology available to date in the market.

Java bean is a simple reusable component. There is no persistency on bean hibernate provides the service to make the bean persistent so that we can make use of it. In hibernate the POJO(Plain java object). represent the DB table.

We are giving the object oriented view using POJO.

13. What is database persistence? How well it is implemented in Hibernate

Ans. 1

Hibernate has an extremely sophisticated granular two-level cache architecture. It is possible to enable or disable the use of the (second-level) process or cluster level cache for a particular class or collection role. There is support for pluggable cache implementations, including EHCache, JBossCache, SwarmCache, Tangosol Coherence Cache. There is also a granular query result set cache. All this flexibility comes at the cost of complexity and can occasionally be tricky for new users.

it will help hibernate to do the persistence work more feasible

14. what is the difference between hibernate and Spring?

Ans. 1.

Hibernate is ORM tool used for data persistency.

Spring is a framework for enterprise applications with default APIs for presentation, middle tiers and persistence layers and allows to integrate with various presentations, middle tier and persistence APIs like Hibernate, Struts, Ibatis, JMS, MQ series etc.

15. what is the use of cascade in hbm file?

Ans. 1.

If cascade=all

It loads all related data

Ans 2.

The use of configuring 'cascade=all' for a particular attribute of a POJO is all the db queries (Save,Update,Delete and Select) against a POJO will always apply to this particular attribute also and no need to execute separately for this attribute.

Ans. 3.

1. cascade="none", the default, tells Hibernate to ignore the association.
- 2) cascade="save-update" tells Hibernate to navigate the association when the transaction is committed and when an object is passed to save() or update() and save newly instantiated transient instances and persist changes to detached instances.
- 3) cascade="delete" tells Hibernate to navigate the association and delete persistent instances when an object is passed to delete().
- 4) cascade="all" means to cascade both save-update and delete, as well as calls to evict and lock.
- 5) cascade="all-delete-orphan" means the same as cascade="all" but, in addition, Hibernate deletes any persistent entity instance that has been removed (dereferenced) from the association (for example, from a collection).
- 6) cascade="delete-orphan" Hibernate will delete any persistent entity instance that has been removed (dereferenced) from the association (for example, from a collection).

Ans. 4.

cascade specifies which operations should be cascaded from the parent object to the associated object. The meaningful values would be persist , merge, delete, save_update, evict , replicate, lock , refresh , all , delete_orphan.

16. How does Hibernate maintain relations of RDBMS?

Ans. 1.

Those are .. 1Â to many, many to many, many to one.. relations.

these relational mappings are done by using the hibernates utilities such as list, set, bags, object...

In detail Go thru....Â www.hibernate.org

17. How are joins handled using HQL.?

Ans. 1.

Its just like RDBMS joining. After joining two or more tables we are retrieving values from different tables. In HQL, The tables are our Value Objects (VOs). when we apply join on two more VOs we got a result list which contains values from different VOs. How to hold these values. There is no VO is available for holding these dynamically created result list.

So, create a POJO class to hold this result. when you are querying form one or more VOs put the result list into the newly created VO.

Eg:

```
select new joinVO (userVO.usercode, UserVO.Username,subjectVo.SubjectName) from UserVO, subjectVO  
where .....
```

Ans. 2.

HQL provides four ways of expressing (inner and outer) joins:

- 1) An ordinary join in the from clause
 - 2) A fetch join in the from clause
 - 3) A theta-style join in the where clause
 - 4) An implicit association join
-

18. What is the main difference between Entity Beans and Hibernate ?

Ans. 1.

- 1)Â In Entity Bean at a time we can interact with only one data Base. Where as in Hibernate we can able to establishes the connections to more than One Data Base. Only thing we need to write one more configuration file.Â Â
-

-
- 2) Entity Beans does not support OOPS concepts where as Hibernate does.
 - 3) Hibernate supports multi level cacheing, where as Entity Beans doesn't.
-

19. What is the difference between hibernate and spring JDBC template? List any advantages and disadvantages

Ans 1.

Spring provides hibernate template and it has many advantages like

- 1) It removes boiler plate code like getting connection from data source, try/catch block for closing connection. So that developer can focus on writing business logic rather than writing boiler plate code everywhere.
 - 2) Spring hibernateTemplate also throws RunTime exception compared to checked exception which allows to remove writing try/catch block in each DAO.
 - 3) It also gives richer template class, using which developer can write query code easily. This template class also allows to get session explicitly, so if developer wants to get session object and work on it, then it's possible.
-

20. What is Hibernate proxy?

Ans. 1.

Class can be mapped to a proxy instead of a table. When you actually call load on session it returns you proxy. This proxy may contain actual method to load the data.

May be used when you already have some pre-existing code for persistence and you want to call it via Hibernate

Ans. 2.

By default Hibernate creates a proxy for each of the class you map in mapping file. This class contains the code to invoke JDBC. This class is created by hibernate using CGLIB

21.What is the difference between and merge and update?

Ans. 1.

In COBOL merging means combining records of two or more sequential files. This can be achieved through MERGE VERB. After merging all the records can be stored in separate output file. In output file all the records are sorted.

Update means we can modify the records of any file. This can be achieved through REWRITE VERB.

22.Why Hibernate is advantageous over Entity Beans & JDBC?

Ans. 1.

Hibernate and EJB have different purpose. EJB is for writing business logic as well and provide Database independency just as hibernate provides for us. EJB in fact uses the container services (EJB container) like transaction, security etc as well. Which is preferred over Hibernate and EJB is actually depend on the application requirement.

EJB has greater complexity than Hibernate. But in many heavily functional web applications it is preferred than hibernate. Other than this you can use EJB where EJB container is present as in case of tomcat it is not present yet. So many web applications which use free tomcat server use hibernate for database independency.

23.What J2EE design problems does Hibernate solves apart from Data Base in-dependency and being an ORM tool?

Ans. 1.

The persistency complexity is migrated from DAO design pattern we just access pojo for the data
