

JavaEra.com[®]

A Perfect Place For Java Learning and Certification

**CoreJava | Jsp | Servlets | JDBC | Struts | Hibernate | Spring
Projects | FAQs | Sample Resumes | eBooks | SCJP | SCWCD
Certification Stuff | Questions Banks | Tutorials | Sample Resumes
Interview Tips | Communities | Forums | Discussions**

www.JavaEra.com

1.What is ORM ?

ORM stands for object/relational mapping. ORM is the automated persistence of objects in a [Java application](#) to the tables in a relational database.

2.What does ORM consists of ?

An ORM solution consists of the followig four pieces:

- API for performing basic CRUD operations
- API to express queries refering to classes
- Facilities to specify metadata
- Optimization facilities : dirty checking,lazy associations fetching
-

3.What are the ORM levels ?

The ORM levels are:

- Pure relational (stored procedure.)
- Light objects mapping (JDBC)
- Medium object mapping
- Full object Mapping (composition,inheritance, polymorphism, persistence by reachability)

4.What is Hibernate?

Hibernate is a pure [Java](#) object-relational mapping (ORM) and persistence framework that allows you to map plain old Java objects to relational database tables using (XML) configuration files.Its purpose is to relieve the developer from a significant amount of relational data persistence-related programming tasks.

5.Why do you need ORM tools like hibernate?

The main advantage of ORM like hibernate is that it shields developers from messy SQL. Apart from this, ORM provides following benefits:

- **Improved productivity**
 - High-level object-oriented API
 - Less Java code to write
 - No SQL to write
- **Improved performance**
 - Sophisticated caching
 - Lazy loading

- Eager loading
- **Improved maintainability**
 - A lot less code to write
- **Improved portability**
 - ORM framework generates database-specific SQL for you

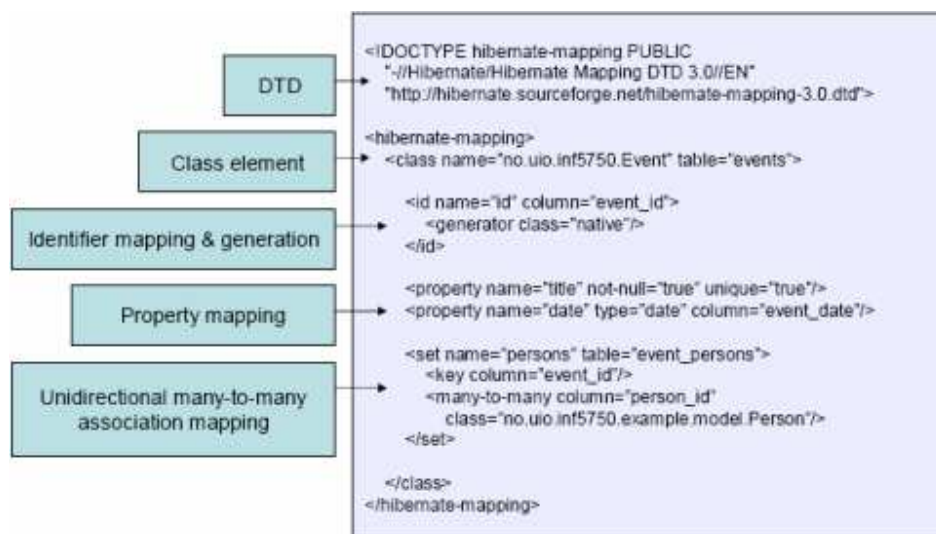
6.What Does Hibernate Simplify?

Hibernate simplifies:

- Saving and retrieving your domain objects
- Making database column and table name changes
- Centralizing pre save and post retrieve logic
- Complex joins for retrieving related items
- Schema creation from object model

7.What is the need for Hibernate xml mapping file?

Hibernate mapping file tells Hibernate which tables and columns to use to load and store objects. Typical mapping file look as follows:



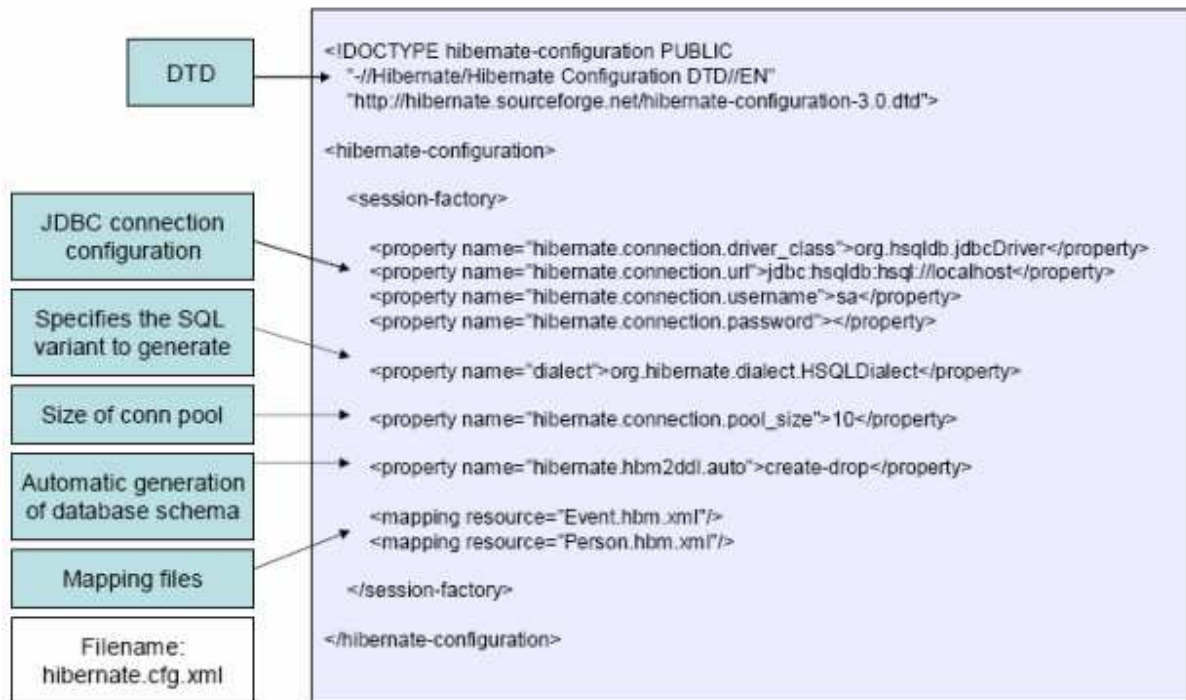
8.What are the most common methods of Hibernate configuration?

The most common methods of Hibernate configuration are:

- Programmatic configuration
- XML configuration (`hibernate.cfg.xml`)

9.What are the important tags of hibernate.cfg.xml?

Following are the important tags of hibernate.cfg.xml:



10. What are the Core interfaces of the Hibernate framework?

The five core interfaces are used in just about every Hibernate [application](#). Using these interfaces, you can store and retrieve persistent objects and control transactions.

- Session interface
- SessionFactory interface
- Configuration interface
- Transaction interface
- Query and Criteria interfaces

11. What role does the Session interface play in Hibernate?

The Session interface is the primary interface used by Hibernate applications. It is a single-threaded, short-lived object representing a conversation between the application and the persistent store. It allows you to create query objects to retrieve persistent objects.

```
Session session = sessionFactory.openSession();
```

Session interface role:

- Wraps a JDBC connection

- Factory for Transaction
- Holds a mandatory (first-level) cache of persistent objects, used when navigating the object graph or looking up objects by identifier

12.What role does the SessionFactory interface play in Hibernate?

The application obtains Session instances from a SessionFactory. There is typically a single SessionFactory for the whole application—created during application initialization. The SessionFactory caches generate SQL statements and other mapping metadata that Hibernate uses at runtime. It also holds cached data that has been read in one unit of work and may be reused in a future unit of work

```
SessionFactory sessionFactory = configuration.buildSessionFactory();
```

13.What is the general flow of Hibernate communication with RDBMS?

The general flow of Hibernate communication with RDBMS is :

- Load the Hibernate configuration file and create configuration object. It will automatically load all hbm mapping files
- Create session factory from configuration object
- Get one session from this session factory
- Create HQL Query
- Execute query to get list containing Java objects

14.What is Hibernate Query Language (HQL)?

Hibernate offers a query language that embodies a very powerful and flexible mechanism to query, store, update, and retrieve objects from a database. This language, the Hibernate query Language (HQL), is an object-oriented extension to SQL.

15.How do you map Java Objects with Database tables?

- First we need to write Java domain objects (beans with setter and getter). The variables should be same as database columns.
- Write hbm.xml, where we map [java class](#) to table and database columns to Java class variables.

Example :

```
<hibernate-mapping>
  <class name="com.test.User" table="user">
    <property column="USER_NAME" length="255"
```

```
        name="userName" not-null="true" type="java.lang.String"/>
    <property column="USER_PASSWORD" length="255"
        name="userPassword" not-null="true" type="java.lang.String"/>
    </class>
</hibernate-mapping>
```

16.What's the difference between load() and get()?

load() vs. get() :-

load()	get()
Only use the load() method if you are sure that the object exists.	If you are not sure that the object exists, then use one of the get() methods.
load() method will throw an exception if the unique id is not found in the database.	get() method will return null if the unique id is not found in the database.
load() just returns a proxy by default and database won't be hit until the proxy is first invoked.	get() will hit the database immediately.

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

Use update() if you are sure that the session does not contain an already persistent instance with the same identifier, and merge() if you want to merge your modifications at any time without consideration of the state of the session.

18.How do you define sequence generated primary key in hibernate?

Using <generator> tag.

Example:-

```
<id column="USER_ID" name="id" type="java.lang.Long">
    <generator class="sequence">
        <param name="table">SEQUENCE_NAME</param>
    </generator>
</id>
```

19.Define cascade and inverse option in one-many mapping?

cascade - enable operations to cascade to child entities.

cascade="all|none|save-update|delete|all-delete-orphan"

inverse - mark this collection as the "inverse" end of a bidirectional association.

inverse="true|false"

Essentially "inverse" indicates which end of a relationship should be ignored, so when persisting a parent who has a collection of children, should you ask the parent for its list of children, or ask the children who the parents are?

20.What do you mean by Named – SQL query?

Named SQL queries are defined in the mapping xml document and called wherever required.

Example:

```
<sql-query name = "empdetails">
  <return alias="emp" class="com.test.Employee"/>
  SELECT emp.EMP_ID AS {emp.empid},
         emp.EMP_ADDRESS AS {emp.address},
         emp.EMP_NAME AS {emp.name}
  FROM Employee EMP WHERE emp.NAME LIKE :name
</sql-query>
```

Invoke Named Query :

```
List people = session.getNamedQuery("empdetails")
                  .setString("TomBrady", name)
                  .setMaxResults(50)
                  .list();
```

21.How do you invoke Stored Procedures?

```
<sql-query name="selectAllEmployees_SP" callable="true">
  <return alias="emp" class="employee">
    <return-property name="empid" column="EMP_ID"/>

    <return-property name="name" column="EMP_NAME"/>
    <return-property name="address" column="EMP_ADDRESS"/>
    { ? = call selectAllEmployees() }
  </return>
</sql-query>
```

22.Explain Criteria API

Criteria is a simplified API for retrieving entities by composing Criterion objects. This is a very convenient approach for functionality like "search" screens where there is a

variable number of conditions to be placed upon the result set.

Example :

```
List employees = session.createCriteria(Employee.class)
                        .add(Restrictions.like("name", "a%"))
                        .add(Restrictions.like("address", "Boston"))
                        .addOrder(Order.asc("name"))
                        .list();
```

23. Define HibernateTemplate?

org.springframework.orm.hibernate.HibernateTemplate is a helper class which provides different methods for querying/retrieving data from the database. It also converts checked HibernateExceptions into unchecked DataAccessExceptions.

24. What are the benefits does HibernateTemplate provide?

The benefits of HibernateTemplate are :

- HibernateTemplate, a Spring Template class simplifies interactions with Hibernate Session.
- Common functions are simplified to single method calls.
- Sessions are automatically closed.
- Exceptions are automatically caught and converted to runtime exceptions.

25. How do you switch between relational databases without code changes?

Using Hibernate SQL Dialects, we can switch databases. Hibernate will generate appropriate hql queries based on the dialect defined.

26. If you want to see the Hibernate generated [SQL](#) statements on console, what should we do?

In Hibernate configuration file set as follows:

```
<property name="show_sql">true</property>
```

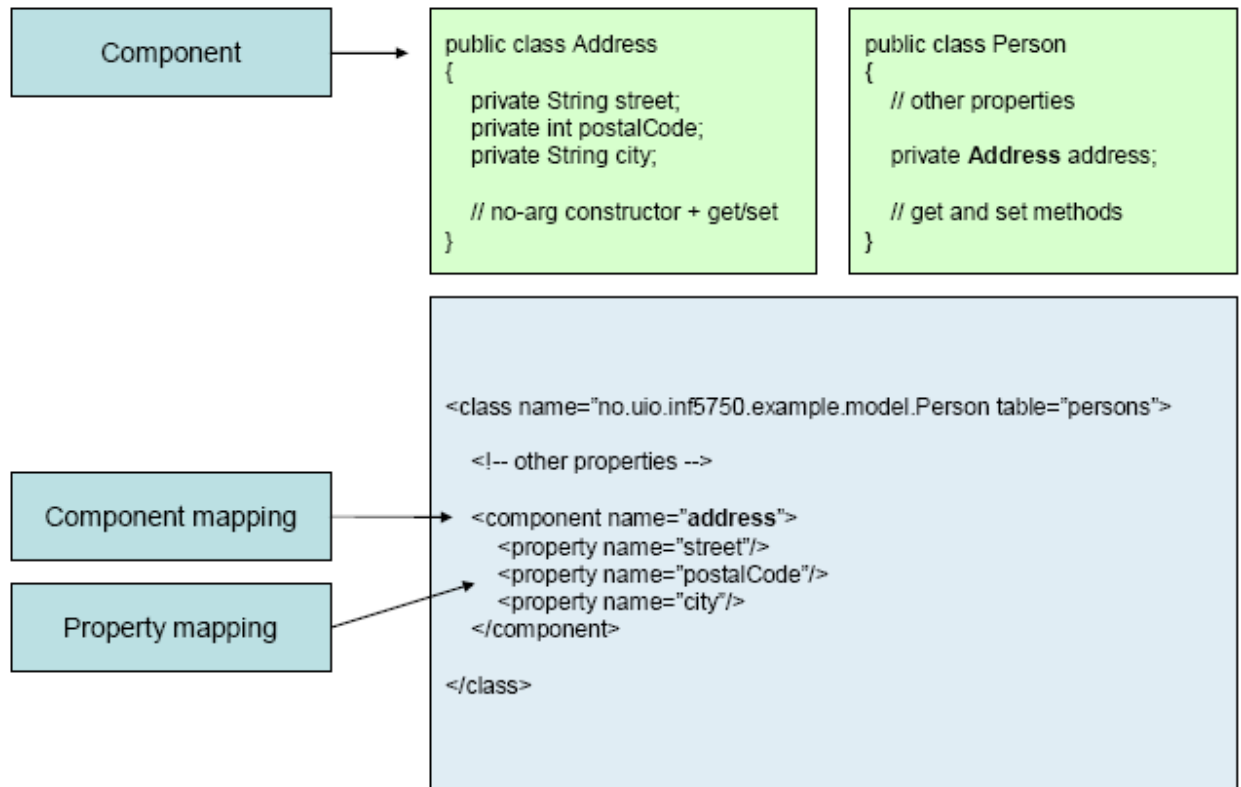
27. What are derived properties?

The properties that are not mapped to a column, but calculated at runtime by evaluation of an expression are called derived properties. The expression can be defined using the formula attribute of the element.

28. What is component mapping in Hibernate?

- A component is an object saved as a value, not as a reference
- A component can be saved directly without needing to declare interfaces or identifier properties
- Required to define an empty constructor
- Shared references not supported

Example:



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

sorted collection vs. order collection :-

sorted collection	order collection
A sorted collection is sorting a collection by utilizing the sorting features provided by the Java collections framework. The sorting occurs in the memory of JVM which running Hibernate, after the data being read from database using java comparator.	Order collection is sorting a collection by specifying the order-by clause for sorting this collection when retrieval.
If your collection is not large, it will be	If your collection is very large, it will be

more efficient way to sort it.

more efficient way to sort it .

31.What is the advantage of Hibernate over jdbc?

Hibernate Vs. JDBC :-

JDBC	Hibernate
With JDBC, developer has to write code to map an object model's data representation to a relational data model and its corresponding database schema.	Hibernate is flexible and powerful ORM solution to map Java classes to database tables. Hibernate itself takes care of this mapping using XML files so developer does not need to write code for this.
With JDBC, the automatic mapping of Java objects with database tables and vice versa conversion is to be taken care of by the developer manually with lines of code.	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.
JDBC supports only native Structured Query Language (SQL). Developer has to find out the efficient way to access database, i.e. to select effective query from a number of queries to perform same task.	Hibernate provides a powerful query language Hibernate Query Language (independent from type of database) that is expressed in a familiar SQL like syntax and includes full support for polymorphic queries. Hibernate also supports native SQL statements. It also selects an effective way to perform a database manipulation task for an application.
Application using JDBC to handle persistent data (database tables) having database specific code in large amount. The code written to map table data to application objects and vice versa is actually to map table fields to object properties. As table changed or database changed then it's essential to change object structure as well as to change code written to map table-to-object/object-to-table.	Hibernate provides this mapping itself. The actual mapping between tables and application objects is done in XML files. If there is change in Database or in any table then the only need to change XML file properties.
With JDBC, it is developer's responsibility to handle JDBC result set and convert it to Java objects through code to use this	Hibernate reduces lines of code by maintaining object-table mapping itself and returns result to application in form of Java

persistent data in application. So with JDBC, mapping between Java objects and database tables is done manually.

With JDBC, caching is maintained by hand-coding.

In JDBC there is no check that always every user has updated data. This check has to be added by the developer.

objects. It relieves programmer from manual handling of persistent data, hence reducing the development time and maintenance cost.

Hibernate, with Transparent Persistence, cache is set to application work space. Relational tuples are moved to this cache as a result of query. It improves performance if client application reads same data many times for same write. Automatic Transparent Persistence allows the developer to concentrate more on business logic rather than this application code.

Hibernate enables developer to define version type field to application, due to this defined field Hibernate updates version field of database table every time relational tuple is updated in form of [Java class](#) object to that table. So if two users retrieve same tuple and then modify it and one user save this modified tuple to database, version is automatically updated for this tuple by Hibernate. When other user tries to save updated tuple to database then it does not allow saving it because this user does not have updated data.

32.What are the Collection types in Hibernate ?

- Bag
- Set
- List
- Array
- Map

33.What are the ways to express joins in HQL?

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

- An *implicit* association join
- An ordinary join in the FROM clause
- A fetch join in the FROM clause.
- A *theta-style* join in the WHERE clause.

34. Define cascade and inverse option in one-many mapping?

cascade - enable operations to cascade to child entities.
cascade="all|none|save-update|delete|all-delete-orphan"

inverse - mark this collection as the "inverse" end of a bidirectional association.
inverse="true|false"

Essentially "inverse" indicates which end of a relationship should be ignored, so when persisting a parent who has a collection of children, should you ask the parent for its list of children, or ask the children who the parents are?

35. What is Hibernate proxy?

The `proxy` attribute enables lazy initialization of persistent instances of the class. Hibernate will initially return CGLIB proxies which implement the named interface. The actual persistent object will be loaded when a method of the proxy is invoked.

36. How can Hibernate be configured to access an instance variable directly and not through a setter method?

By mapping the property with `access="field"` in Hibernate metadata. This forces hibernate to bypass the setter method and access the instance variable directly while initializing a newly loaded object.

37. How can a whole class be mapped as immutable?

Mark the class as `mutable="false"` (Default is true). This specifies that instances of the class are (not) mutable. Immutable classes, may not be updated or deleted by the application.

38. What is the use of `dynamic-insert` and `dynamic-update` attributes in a class mapping?

Criteria is a simplified API for retrieving entities by composing Criterion objects. This is a very convenient approach for functionality like "search" screens where there is a variable number of conditions to be placed upon the result set.

- `dynamic-update` (defaults to `false`): Specifies that `UPDATE SQL` should be generated at runtime and contain only those columns whose values have changed
- `dynamic-insert` (defaults to `false`): Specifies that `INSERT SQL` should be generated at runtime and contain only the columns whose values are not null.

39.What do you mean by fetching strategy ?

A *fetching strategy* is the strategy Hibernate will use for retrieving associated objects if the application needs to navigate the association. Fetch strategies may be declared in the O/R mapping metadata, or over-ridden by a particular HQL or Criteria query.

40.What is automatic dirty checking?

Automatic dirty checking is a feature that saves us the effort of explicitly asking Hibernate to update the database when we modify the state of an object inside a transaction.

41.What is transactional write-behind?

Hibernate uses a sophisticated algorithm to determine an efficient ordering that avoids database foreign key constraint violations but is still sufficiently predictable to the user. This feature is called transactional write-behind.

42.What are Callback interfaces?

Callback interfaces allow the application to receive a notification when something interesting happens to an object—for example, when an object is loaded, saved, or deleted. Hibernate applications don't need to implement these callbacks, but they're useful for implementing certain kinds of generic functionality.

43.What are the types of Hibernate instance states ?

Three types of instance states:

- Transient -The instance is not associated with any persistence context
- Persistent -The instance is associated with a persistence context
- Detached -The instance was associated with a persistence context which has been closed – currently not associated

44.What are the differences between EJB 3.0 & Hibernate

Hibernate Vs EJB 3.0 :-

Hibernate	EJB 3.0
-----------	---------

Session —Cache or collection of loaded objects relating to a single unit of work	Persistence Context —Set of entities that can be managed by a given EntityManager is defined by a persistence unit
XDoclet Annotations used to support Attribute Oriented Programming	Java 5.0 Annotations used to support Attribute Oriented Programming
Defines HQL for expressing queries to the database	Defines EJB QL for expressing queries
Supports Entity Relationships through mapping files and annotations in JavaDoc	Support Entity Relationships through Java 5.0 annotations
Provides a Persistence Manager API exposed via the Session, Query, Criteria, and Transaction API	Provides and Entity Manager Interface for managing CRUD operations for an Entity
Provides callback support through lifecycle, interceptor, and validatable interfaces	Provides callback support through Entity Listener and Callback methods
Entity Relationships are unidirectional. Bidirectional relationships are implemented by two unidirectional relationships	Entity Relationships are bidirectional or unidirectional

45.What are the types of inheritance models in Hibernate?

There are three types of inheritance models in Hibernate:

- Table per class hierarchy
- Table per subclass
- Table per concrete class

Part II

Hibernate Interview Questions

[Hibernate Persistent technology](#)

Can anybody tell me which persistent technology can be used instead of Hibernate which is as compatible as Hibernate (not [JDBC](#) nor [EJB](#)) any parallel technology?

Latest Answer: Use IBatis for persistent, IBatis is Good Object Releation Mapping ...

[Read Answers \(6\)](#) | Asked by : [pravinpawade](#)

[Answer Question](#) [Subscribe](#)

[Stored procedure in hibernate](#)

How to call stored procedure in mysql through hibernate?

Latest Answer: In order to call a stored procedure using the hibernate , define a named query for a persistent class mapping document. Then call the named query from your java application .Example is given in the Hibernate Reference documentation as follows:-First create ...

[Read Answers \(3\)](#) | Asked by : [bhupeshb](#)

[Answer Question](#) [Subscribe](#)

[What is different between Hibernate and iBatis?](#)

Latest Answer: These are the few questions that continuously get asked in most of forums. Whatâ€™s really difference between two andreally more importantly when should I use one over the other. Its pretty interesting question because there are major differences ...

[Read Answers \(3\)](#) | Asked by : [siva2baba](#)

[Answer Question](#) [Subscribe](#)

[Why Hibernate came in to picture](#)

Anybody tell me please, where exactly Hibernate is used..tell me about the mapping and .xml file in Hibernate

Latest Answer: To make the application developemnt more productive. its saves development time and also mainly it deals with Objects(POJO) . and .xml file is nothing but mapping between database column and POJO variable. you can easily switch the database like MySql ...

[Read Answers \(2\)](#) | Asked by : [udayvkumar](#)

[Answer Question](#) [Subscribe](#)

[Lazy initialisation](#)

what is lazy initialisation in hibernate

Latest Answer: by default lazy = true so that parent table wont fetch the fields from child table. so in that case u have to write explicitly lazy = false Then we can fetch the fields from the parent as well as child fields ...

[Read Answers \(5\)](#) | Asked by : [gdkundu](#)

[Answer Question](#) [Subscribe](#)

[Execute Stored procedure in Hibernate](#)

How to invoke a stored procedure in Hibernate and pass in a parameter? I have { ? = call myservice.TEST_PROC3(?) } While I can get stored proc output if there is no input to TEST_PROC3, e.g. call

Latest Answer: You can declare it as Named queries in one of your mapping files and give unique name. Call it in your code. Sample code is below. SQLQuery sq =

([SQLQuery](#))

[session](#).getNamedQuery("findSearchResultsListSP"); sq.addEntity("documentTypeCode",
...

[Read Answers \(4\)](#) | Asked by : newdeveloper

[Answer Question](#) [Subscribe](#)

[What J2EE design problems does Hibernate solve apart from Data Base in-dependency and being an ORM](#)

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

[Read Answers \(2\)](#) | Asked by : Rajeev Ranjan

[Answer Question](#) [Subscribe](#)

[Why Hibernate is advantageous over Entity Beans & JDBC?](#)

Latest Answer: In my opinion in case of [Entity](#) Beans our bean only work under container. means you can't reuse those object outside container. There is no concept of Detached Objects. But in case of hibernate we have detached object concepts. ...

[Read Answers \(1\)](#) | Asked by : gopikrishna

[Answer Question](#) [Subscribe](#)

[What is the difference between merge and update](#)

Latest Answer: Use update() if you are sure that the session does not contain an already persistent instance with the same identifier, and merge() if you want to merge your modifications at any time without consideration of the state of the session. In other words, ...

[Read Answers \(5\)](#) | Asked by : [rajneeshg](#)

[Answer Question](#) [Subscribe](#)

[What is Hibernate proxy?](#)

Latest Answer: Proxies are created dynamically by subclassing your object at runtime. The subclass has all the [methods](#) of the parent, and when any of the methods are accessed, the proxy loads up the real object from the DB and calls the [method](#) for you. Very nice in ...

How will you configure Hibernate?

Answer:

The configuration files hibernate.cfg.xml (or hibernate.properties) and mapping files *.hbm.xml are used by the Configuration class to create (i.e. configure and bootstrap hibernate) the SessionFactory, which in turn creates the Session instances. Session instances are the primary interface for the persistence service.

" hibernate.cfg.xml (alternatively can use hibernate.properties): These two files are used to configure the hibernate service (connection driver class, connection URL, connection username, connection password, dialect etc). If both files are present in the classpath then hibernate.cfg.xml file overrides the settings found in the hibernate.properties file.

" Mapping files (*.hbm.xml): These files are used to map persistent objects to a relational database. It is the best practice to store each object in an individual mapping file (i.e. mapping file per class) because storing large number of persistent classes into one mapping file can be difficult to manage and maintain. The naming convention is to use the same name as the persistent (POJO) class name. For example Account.class will have a mapping file named Account.hbm.xml. Alternatively hibernate annotations can be used as part of your persistent class code instead of the *.hbm.xml files.

Q. What is a SessionFactory? Is it a thread-safe object?

Answer:

SessionFactory is Hibernate's concept of a single datastore and is thread-safe so that many threads can access it concurrently and request for sessions and immutable cache of compiled mappings for a single database. A SessionFactory is usually only built once at startup. SessionFactory should be wrapped in some kind of singleton so that it can be easily accessed in an application code.

```
SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();
```

Q. What is a Session? Can you share a session object between different threads?

Answer:

Session is a light weight and a non-threadsafe object (No, you cannot share it between threads) that represents a single unit-of-work with the database. Sessions are opened by a SessionFactory and then are closed when all work is complete. Session is the primary interface for the persistence service. A session obtains a database connection lazily (i.e. only when required). To avoid creating too many sessions ThreadLocal class can be used as shown below to get the current session no matter how many times you make call to the currentSession() method.

```
&
public class HibernateUtil {
&
public static final ThreadLocal local = new ThreadLocal();

public static Session currentSession() throws HibernateException {
    Session session = (Session) local.get();
    //open a new session if this thread has no session
    if(session == null) {
        session = sessionFactory.openSession();
        local.set(session);
    }
    return session;
}
}
```

It is also vital that you close your session after your unit of work completes. Note: Keep your Hibernate Session API handy.

Q. What are the benefits of detached objects?

Answer:

Detached objects can be passed across layers all the way up to the presentation layer without having to use any DTOs (Data Transfer Objects). You can later on re-attach the detached objects to another session.

Q. What are the pros and cons of detached objects?

Answer:

Pros:

" When long transactions are required due to user think-time, it is the best practice to break the long transaction up into two or more transactions. You can use detached objects from the first transaction to carry data all the way up to the presentation layer. These detached objects get modified outside a transaction and later on re-attached to a new transaction via another session.

Cons

" In general, working with detached objects is quite cumbersome, and better to not clutter up the session with them if possible. It is better to discard them and re-fetch them on subsequent requests. This approach is not only more portable but also more efficient because - the objects hang around in Hibernate's cache anyway.

" Also from pure rich domain driven design perspective it is recommended to use DTOs (DataTransferObjects) and DOs (DomainObjects) to maintain the separation between Service and UI tiers.

Q. How does Hibernate distinguish between transient (i.e. newly instantiated) and detached objects?

Answer

" Hibernate uses the `version` property, if there is one.

" If not uses the identifier value. No identifier value means a new object. This does work only for Hibernate managed surrogate keys. Does not work for natural keys and assigned (i.e. not managed by Hibernate) surrogate keys.

" Write your own strategy with `Interceptor.isUnsaved()`.

Q. What is the difference between the `session.get()` method and the `session.load()` method?

Both the `session.get(..)` and `session.load()` methods create a persistent object by loading the required object from the database. But if there was not such object in the database then the method `session.load(..)` throws an exception whereas `session.get(&)` returns null.

Q. What is the difference between the `session.update()` method and the `session.lock()` method?

Both of these methods and `saveOrUpdate()` method are intended for reattaching a detached object. The `session.lock()` method simply reattaches the object to the session without checking or updating the database on the assumption that the database is sync with the detached object. It is the best practice to use either `session.update(..)` or `session.saveOrUpdate()`. Use `session.lock()` only if you are absolutely sure that the detached object is in sync with your detached object or if it does not matter because you will be overwriting all the columns that would have changed later on within the same transaction.

Note: When you reattach detached objects you need to make sure that the dependent objects are reached as well.

Q. How would you reattach detached objects to a session when the same object has already been loaded into the session?

You can use the `session.merge()` method call.

Q. What are the general considerations or best practices for defining your Hibernate persistent classes?

1. You must have a default no-argument constructor for your persistent classes and there should be `getXXX()` (i.e. accessor/getter) and `setXXX()` (i.e. mutator/setter) methods for all your persistable instance variables.
2. You should implement the `equals()` and `hashCode()` methods based on your business key and it is important not to use the id field in your `equals()` and `hashCode()` definition if the id field is a surrogate key (i.e. Hibernate managed identifier). This is because the Hibernate only generates and sets the field when saving the object.
3. It is recommended to implement the `Serializable` interface. This is potentially useful if you want to migrate around a multi-processor cluster.
4. The persistent class should not be final because if it is final then lazy loading cannot be used by creating proxy objects.
5. Use XDoclet tags for generating your `*.hbm.xml` files or Annotations (JDK 1.5 onwards), which are less verbose than `*.hbm.xml` files.

Hibernate Interview Questions

1) Explain about Hibernate?

Hibernate solves problems such as Object Relational impedance mismatch, etc. It is commonly used for object and query service. It helps data base developers develop classes which include inheritance, association, composition and polymorphism. A developer or user can express queries either in HQL or SQL.

2) Explain about the primary feature of Hibernate?

Primary feature of hibernate is to java classes to database tables. Data query and retrieval is also possible with Hibernate. Application portability is a key feature in Hibernate it allows developers to port applications to almost all SQL databases.

3) Explain about transparent persistence of Hibernate?

Transparent persistence is provided for Plain old Java objects or POJOs. For proper functioning of the applications importance should be given to the methods equals () and hash Code methods (). It has a requirement which should be strictly followed in the applications which is a no-argument constructor.

4) Explain about the dirty checking feature of Hibernate?

Dirty checking feature of the Hibernate allows users or developers to avoid time consuming data base write actions. This feature makes necessary updations and changes to the fields which require a change, remaining fields are left unchanged or untouched.

5) Brief about the Session factory interface?

It creates new hibernate sessions by referencing immutable and thread safe objects. Application using hibernate are usually allowed and desgined to implement single instance of the class using this interface. Only single instance of a class can be used which is using this interface.

6) Explain about session interface?

This represents hibernate session which perform the manipulation on the database entities. Some of the activities performed by session interface are as follows they are managing the persistence state, fetching persisted ones and management of the transaction demarcation.

7) Explain the steps involved in creating database applications with Java using Hibernate?

Creating Database applications with Java is made simpler with Hibernate. First Plain old java object needs to be written, XML mapping file should be created which shows relationship between database and class attributes. Hibernate APIs can be used to store persistent objects.

8) Explain about hibernate.cfg.xml?

Hibernate can be configured with two types of files out of which hibernate.cfg.xml is widely used and popular feature. Hibernate consults hibernate.cfg.xml file for its operating properties such as database dialect, connection string and mapping files. These files are searched on class path.

9) Explain about mapping description file?

Mapping description file is the second file which Hibernate uses to configure its functions. This mapping file has an extension *.hbm which instructs mapping between Java class and database tables. The usage of mapping description file rests entirely upon the business entity.

10) Explain about transaction file?

Transactions denote a work file which can save changes made or revert back the changes. A transaction can be started by session.beginTransaction() and it uses JDBC connection, CORBA or JTA. When this session starts several transactions may occur.

11) Explain about mapping files in Hibernate?

Mapping files forms the core of any database mapping tools. These files contain field to field mapping, usually this mapping occurs between classes and attributes. After mapping files they can be persist to the database. Tags can be used to indicate the presence of a primary key.

12) What is the effect when a transient mapped object is passed onto a Sessions save?

When a Sessions save () is passed to a transient mapped object it makes the method to become more persistent. Garbage collection and termination of the Java virtual machine stays as long as it is deleted explicitly. It may head back to its transient state.

13) Explain about version field?

Application level data integrity constants are important if you are making changes to offline information which is again backed by database. Higher level locking or versioning protocol is required to support them. Version field usage comes at this stage but the design and implementation process is left to the developer.

14) State some advantages of hibernate?

Some of the advantages which a developer can get from Hibernate are as follows: -

Mapping of one POJO table to one table is not required in hibernate.

It supports inheritance relationships and is generally a fast tool. Portability is necessary the greater benefit from hibernate. POJOs can be used in other applications where they are applicable.

15) Explain about addClass function?

This function translates a Java class name into file name. This translated file name is then loaded as an input stream from the Java class loader. This addclass function is important if you want efficient usage of classes in your code.

16) Explain about addjar() and addDirectory() methods?

These methods are the most convenient to use in hibernate. These methods allow you to load all your Hibernate documents at a time. These methods simplify code configuration, refactoring, layout, etc. These functions help you to add your hibernate mapping to Hibernate initialization files.

17) Explain about the id field?

This id field corresponds to the surrogate key which is generated by the database. These fields are handled by the id field. Name attribute is used to specify the names of the field and it should correspond to the method name of getid. This also should correspond to long type and the values should be stored in the database in the long column.

Can the strategy of LAZY load for the dependent domain classes, be change once SessionFactory is created?

Answer to the Best of my knowledge:

No, as SessionFactory is loaded/created once for the entire lifecycle of the application, and while creating Hibernate SessionFactory, appropriate HBM XML file is read and these configuration are used for creating or instantiating Hibernate SessionFactory, so it is logically not possible as, SessionFactory is not going to refresh but destroyed and recreated. And for different modules from the same application, if requires LAZY as true / false, then using a single Hibernate SessionFactory won't be able to serve both these Persistent Object loading strategies.

Hibernate Question :

What are the different Transaction Factories available with Hibernate?

Hibernate Answer :

There are three different types of Transaction Factory available with Hibernate 3.2 as JDBCTransactionFactory, JTATransactionFactory and CMTTransactionFactory.

Hibernate Question :

Which one is the default transaction factory in Hibernate 3.2?

Hibernate interview answer

JDBCTransactionFactory is the default local transaction factory with Hibernate 3.2.

Hibernate interview question

Can Hibernate Session Factory be bound to JNDI?

Hibernate interview answer

Yes, by configuring in hibernate.cfg file, session factory can be bound to initial context (as defined by properties hibernate.jndi.url

and hibernate.jndi.class).

Hibernate interview question

Can Hibernate be used to call stored procedures and SQL statements?

Hibernate interview answer

Yes, there are provision in Hibernate 3.2, for defining callable statements and SQL in mapping HBM files.

Hibernate interview question

Can the custom SQL be defined for creation of Java entity object by loading values from database tables and populating Java Object?

Hibernate interview answer

Yes, Javaentity objects can be loaded with custom SQL queries and can be defined in HBM file in form of HQL (Hibernate Query Language).

Hibernate interview question

What are the different Fetching Strategies available with Hibernate 3.2?

Hibernate interview answer

There are four different Fetching standards available in Hibernate3.2, as follows: join fetching, select fetching, batch fetching, sub-select fetching.

Hibernate interview question

What are the different types of statistics available in Hibernate 3.2?

Hibernate interview answer

Different types of statistics like QueryStatistics, CategorizedStatistics, CollectionStatistics, EntityStatistics etc., available in Hibernate 3.2.

Hibernate interview question

How can you get a handle on Hibernate Statistics?

Hibernate interview answer

If Hibernate is deployed in a JMX enabled Application server, then Hibernate provided a statistics service, that can be registered as MBean with JMX server and be used to retrieve different types of statistics available. Hibernate statistics can be obtained from session factory as well.

Hibernate interview question

Can Hibernate be used to map persistent entity POJO to XML files?

Hibernate interview answer

Yes, Hibernate can be used to mapp XML file/tags to POJO entity classes.

Hibernate Question : If there are multiple databases to be used to interact with domain classes, how can session factory be able to manage multiple datasources?

Hibernate Answer :

Each datasource will be configured to each session factory, and to use a single database, a session is created to use database.

Question : What is lazy initialization in Hibernate?

Answer :

When there is an association of one-to-one, or one-to-many, or many-to-many between classes, and on creation of one object, it has to be decided whether to bring associated objects along with this object or not. By setting lazy="true" we instruct Hibernate not to bring the associated object/objects during creation of the required object. By setting lazy="false", it is the reverse, this means we instruct Hibernate to bring all the associated objects also at the time of returning the associating object.

Hibernate interview Question : if there any impact on performance by this attribute lazy ?

Hibernate interview Answer :

This is purely a configuration time decision one has to take to use lazy attribute and its value (true/false) appropriately. As SessionFactory is created once and reused, all the configuration setting in HBM file is read once, and can't be changed at runtime.

Hibernate Question : What are the different states of an instance in Hibernate?

Hibernate Answer :

There are three states that exist for any instance of a class. These are transient, persistent and detached. Those instances that are created but not associated with any session or not saved in database are transient objects. Those instances that are created and be used in any of the methods like save, saveOrUpdate, update of Session are called persistent objects. Those instances that were used in Session methods like save, saveOrUpdate or update to be inserted or updated in database table, and then session is flushed and closed, now these objects are in JVM, but these are not bound to any session.

Hibernate interview question

How can certain type of logic executed on execution of CRUD operation of session, without duplicating it across many places in code base?

Hibernate interview answer

Hibernate Interceptors can be used to receive callback for certain type of events or operations like save, delete, load, update of session. Session Factory level interceptor and session level interceptor. These Interceptors can be used to have code for certain type of logic to be called for every lifecycle method of session.

Hibernate interview question

How can multiple threads access session factory simultaneously to create session instance?

Hibernate interview answer

session factory is thread-safe, so it is okay to be used by many threads to have session from session factory, but I think session is not thread safe and it should be used by one thread at a time, and after use, session has to be flushed and closed.

Hibernate interview question

How many ways Hibernate manages concurrency ?

Hibernate interview answer

Hibernate has different ways of managing concurrency. These are automatic versioning, detached object and extended user sessions.

Hibernate interview question

What is the difference between uni-directional and bi-directional associations?

Hibernate interview answer

uni-directional association allows object creation from one direction only. Bi-directional association allows object querying from both directions of fetching object instances.

A->B, now querying A, can provide information on B as well, based on lazy parameter, but in case of A<->B, querying either A or B, will have value of B or A as well, respectively.

Hibernate interview Question

What are the different contextual session in Hibernate?

Hibernate interview answer

There are three different types of contextual session Hibernate provides, these are JTA session context, local thread session context and managed session context. JTA session context is applicable in case Hibernate

session is running in JTA (Java Transaction API), request thread level session scoped applicable in case of local thread session, and managed session, requires application to open, close and flush session, so creation of session should be handled by application only.

Hibernate interview Question

Can you tell us difference between Hibernate HQL over SQL?

Hibernate interview answer

HQL is fully object oriented, with support for object inheritance, polymorphism and association, but SQL is more of Relational with structured form of queries.

Hibernate interview Question

What are the different scopes one can introduce while using Interceptors with Hibernate?

Hibernate interview Answer

Probably, one can use interceptors with hibernate Session scoped or SessionFactory scoped contexts, while using Interceptors with Hibernate.

Hibernate interview Question

How many ways client application that uses Hibernate to react to certain events?

Hibernate interview Answer

Probably, if I am not wrong, two ways one can react/act to certain events generated out of Hibernate Framework. These are either Interceptors or event systems.

Hibernate interview Question

Can I be able to persist a XML DOM object tree to database by defining mapping between XML DOM to database table, without using POJOs?

Hibernate interview Answer

Yes, one can use Hibernate mapping to persist XML DOM tree hierarchy to database tables.

Hibernate Interview Questions

[What is the difference between hibernate and spring JDBC template? List any advantages and disadvantages](#)

What is the difference between hibernate and spring [JDBC](#) template? List any advantages and disadvantages

[Read Answers \(2\)](#) | Asked by : kasim

[Answer Question](#) [Subscribe](#)

[What is the main difference between Entity Beans and Hibernate ?](#)

Latest Answer: The main difference between [Entity](#) Bean and hibernate is Entity Bean is a heavy weight component and hibernate is a light weight component. ...

[Read Answers \(3\)](#) | Asked by : rajeah

[Answer Question](#) [Subscribe](#)

[How are joins handled using HQL.](#)

Latest Answer: HQL provides four ways of [expressing](#) (inner and outer) joins:1) An ordinary join in the from clause2) A fetch join in the from clause3) A theta-style join in the where clause4) An implicit association join ...

[Read Answers \(2\)](#) | Asked by : Sam

[Answer Question](#) [Subscribe](#)

[How does Hibernate maintain relations of RDBMS](#)

Latest Answer: Those are .. 1) one 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.... 2) www.hibernate.org ...

[Read Answers \(1\)](#) | Asked by : M Anand kumar

Tags : [RDBMS](#)

[Answer Question](#) [Subscribe](#)

[How to use hibernate in NetBeans IDE?](#)

Latest Answer: Add the hibernate related jar files into lib directory of any web application using net beans.<http://www.netbeans.org/kb/41/hibernate.html> ...

[Read Answers \(1\)](#) | Asked by : Hem

[Answer Question](#) [Subscribe](#)

[What is the use of cascade in hbm file?](#)

Latest Answer: 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. ...

[Read Answers \(4\)](#) | Asked by : ravi

[Answer Question](#) [Subscribe](#)

[What is the difference between hibernate and Spring](#)

Latest Answer: 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 ...

[Read Answers \(1\)](#) | Asked by : HARSHAL

[Answer Question](#) [Subscribe](#)

[What is database persistence?How well it is implemented in Hibernate](#)

Latest Answer: Preserving the data inside a database is database persistence. Persistence is one of the fundamental [concepts](#) of application development. There are many [alternatives](#) to implement persistence. Object/relational mapping (ORM) is the technique that Hibernate ...

[Read Answers \(2\)](#) | Asked by : donna

[Answer Question](#) [Subscribe](#)

[What is the difference between beans and hibernate\(which one is good\)?](#)

Latest Answer: Its is 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 ...

[Read Answers \(4\)](#) | Asked by : [enjoy](#)

[Answer Question](#) [Subscribe](#)

[What is the latest version of Hibernate using in current industry?](#)

Latest Answer: Hibernate 3.2 ...

[Read Answers \(2\)](#) | Asked by : [enjoy](#)

Hibernate Interview Questions

[Q1: What is the difference between Hibernate and EJB 2.1? Q2: Which one, among EJB 2.1 and Hibernate,](#)

[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.](#)

[Read Answers \(2\)](#) | Asked by : Kumar

[Answer Question](#) [Subscribe](#)

[I am using hibernate.What is the use In Data\(POJO\) class primary key is sending as an argument with](#)

I am using hibernate.What is the use In Data(POJO) class primary key is sending as an argument with constructor?

[View Question](#) | Asked by : Kotireddy

[Answer Question](#) [Subscribe](#)

[What is the advantage of Hibernate over jdbc?](#)

Latest Answer: 1) Hiberante is not tightly tied with any underlaying database.Where as [JDBC](#) is tightly tied with the underlaying database."Write Once presist anywhere" using hibernate can achieved by changing the dialect in configuration [xml](#) file.where as ...

[Read Answers \(6\)](#) | Asked by : kounteya

[Answer Question](#) [Subscribe](#)

[I am new to Hibernate.So can any one tell me how to develop a application using Hibernate in the Myeclipse](#)

I am new to Hibernate.So can any one tell me how to develop a application using Hibernate in the Myeclipse .and also please send me pdf to suddu026@gmail.com.Regards,Sudarshan.

[Read Answers \(3\)](#) | Asked by : Sudarshan..

[Answer Question](#) [Subscribe](#)

[How to create primary key using hibernate?](#)

Latest Answer: Ccreate primary key? Not sure but you can defined it in the hbm file by specifying the field with the primary column in the db. ...

[Read Answers \(5\)](#) | Asked by : satyanarayana

[Answer Question](#) [Subscribe](#)

[What is the main advantage of using the hibernate than using the sql](#)

Latest Answer: Hibernate is based on object oriented concept like [java](#). so it has better compatibility with java than sql.In Jdbc we have to mannualy handle exception and every time we need to open/cose the connection,create/close the statement , resultset whatever ...

[Read Answers \(7\)](#) | Asked by : davis

[Answer Question](#) [Subscribe](#)

[What is the difference between sorted and orderd collection in hibernate?](#)

Latest Answer: A component is an object saved as a value, not as a reference A component can be saved directly without needing to declare [interfaces](#) or identifier properties Required to define an empty constructor Shared references not supported ...

[Read Answers \(3\)](#) | Asked by : [sunils](#)

[Answer Question](#) [Subscribe](#)

[Why do you need ORM tools like hibernate?](#)

Latest Answer: 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 a large amount of data have to handle that time instead of using JDBC we used Hibernate. ...

[Read Answers \(5\)](#) | Asked by : [sunils](#)

[Answer Question](#) [Subscribe](#)

[How do you handle the situation where persistence class attribute name is one of the reserved keyword](#)

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?

[Read Answers \(4\)](#) | Asked by : [sunils](#)

[Answer Question](#) [Subscribe](#)

[What is component mapping in hibernate?](#)

Latest Answer: 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{chat initial;String ...

Hibernate Interview Questions

[How JPA and hibernate are related as per EJB 3 specifications?](#)

Latest Answer: first things first ..i have just seen ppl venting thier personal grudge over [java](#) / jpa or whatever over this forum.suns official sacceptence that jpa is a failure. please lets keep it out.as for the answer , JPA is a specification , its nt a frame work ...

[Read Answers \(5\)](#) | Asked by : [sunils](#)

[Answer Question](#) [Subscribe](#)

[What is lazy fetching in hibernate](#)

Latest Answer: Lazy loading is the responsibility is that to load the objects for its parentIn mapping file, if you want to load the child objects while loading the parent it will wont load coz by default load = true so you have to specify lazy = false here child ...

[Read Answers \(6\)](#) | Asked by : j2eeinterviewee

[Answer Question](#) [Subscribe](#)

[What are the types of inheritance models and describe how they work like vertical inheritance and horizontal](#)

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