

# Hibernate Job Interview Questions And Answers



**Interview Questions Answers**

<https://interviewquestionsanswers.org/>

## About Interview Questions Answers

**Interview Questions Answers . ORG** is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Hibernate will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Hibernate Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Hibernate category. To ensure quality, each submission is checked by our team, before it becomes live. This [Hibernate Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.  
[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter for latest Jobs and interview preparation guides.  
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

**Interview Questions Answers.ORG Team**  
[https://InterviewQuestionsAnswers.ORG/  
Support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG)



## Hibernate Interview Questions And Answers Guide.

### Question - 1:

How to avoid constraint violation exception parent key not found?

#### Ans:

No Answer is Posted For this Question

Be First To [Post Your Answer Now.](#)

### Question - 2:

What are the ORM level?

#### Ans:

The ORM levels are:

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

[View All Answers](#)

### Question - 3:

What does ORM consists of?

#### Ans:

An ORM solution consists of the following four pieces:

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

[View All Answers](#)

### Question - 4:

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

#### Ans:

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 persistent 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.

[View All Answers](#)

### Question - 5:

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

#### Ans:

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

[View All Answers](#)

### Question - 6:

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

#### Ans:



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 in 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 reattached as well.

[View All Answers](#)

### Question - 7:

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

#### Ans:

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.

[View All Answers](#)

### Question - 8:

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

#### Ans:

" 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()`.

[View All Answers](#)

### Question - 9:

What are the pros and cons of detached objects?

#### Ans:

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.

[View All Answers](#)

### Question - 10:

What are the benefits of detached objects?

#### Ans:

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.

[View All Answers](#)

### Question - 11:

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

#### Ans:

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.

[View All Answers](#)

### Question - 12:

What is a `SessionFactory`? Is it a thread-safe object?

#### Ans:

`SessionFactory` is Hibernate's concept of a single datastore and is threadsafe 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.

[View All Answers](#)

### Question - 13:

How will you configure Hibernate?

#### Ans:

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 sevice (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.

[View All Answers](#)

### Question - 14:

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

#### Ans:

sorted collection vs. order collection

sorted 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.

If your collection is not large, it will be more efficient way to sort it.

order collection :-

Order collection is sorting a collection by specifying the order-by clause for sorting this collection when retrieval.

If your collection is very large, it will be more efficient way to sort it .

[View All Answers](#)

### Question - 15:

What is component mapping in Hibernate?

#### Ans:

\* 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

[View All Answers](#)

### Question - 16:

What are derived properties?

#### Ans:

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.

People who read this also read:

Core Java Questions

Spring Questions

SCJP 6.0 Certification

EJB Interview Questions

Servlets Questions

[View All Answers](#)

### Question - 17:

If you want to see the Hibernate generated SQL statements on console, what should we do?

#### Ans:

In Hibernate configuration file set as follows:

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

[View All Answers](#)

### Question - 18:

How do you switch between relational databases without code changes?

#### Ans:

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

[View All Answers](#)

### Question - 19:

What are the benefits does HibernateTemplate provide?

#### Ans:

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.

[View All Answers](#)

### Question - 20:



Define HibernateTemplate?

**Ans:**

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.

[View All Answers](#)

**Question - 21:**

Explain Criteria API?

**Ans:**

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();
```

[View All Answers](#)

**Question - 22:**

How do you invoke Stored Procedures?

**Ans:**

```
<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>
```

[View All Answers](#)

**Question - 23:**

What do you mean by Named - SQL query?

**Ans:**

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();
```

[View All Answers](#)

**Question - 24:**

What does it mean to be inverse?

**Ans:**

It informs hibernate to ignore that end of the relationship. If the one-to-many was marked as inverse, hibernate would create a child->parent relationship (child.getParent). If the one-to-many was marked as non-inverse then a child->parent relationship would be created.

[View All Answers](#)

**Question - 25:**

Define cascade and inverse option in one-many mapping?

**Ans:**

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?

[View All Answers](#)



### Question - 26:

How do you define sequence generated primary key in hibernate?

#### Ans:

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>
```

[View All Answers](#)

### Question - 27:

What is the difference between and merge and update?

#### Ans:

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.

[View All Answers](#)

### Question - 28:

What is the difference between load() and get()?

#### Ans:

load() vs. get()

load() :-

Only use the load() method if you are sure that the object exists.

load() method will throw an exception 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():-

If you are not sure that the object exists, then use one of the get() methods.

get() method will return null if the unique id is not found in the database.

get() will hit the database immediately.

[View All Answers](#)

### Question - 29:

What Does Hibernate Simplify?

#### Ans:

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

[View All Answers](#)

### Question - 30:

How do you map Java Objects with Database tables?

#### Ans:

\* 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>
```

[View All Answers](#)

### Question - 31:

What is Hibernate Query Language (HQL)?

#### Ans:

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.

[View All Answers](#)

### Question - 32:



What is the general flow of Hibernate communication with RDBMS?

**Ans:**

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

[View All Answers](#)

**Question - 33:**

What role does the SessionFactory interface play in Hibernate?

**Ans:**

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();

[View All Answers](#)

**Question - 34:**

What role does the Session interface play in Hibernate?

**Ans:**

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

[View All Answers](#)

**Question - 35:**

What are the Core interfaces are of Hibernate framework?

**Ans:**

People who read this also read:

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

[View All Answers](#)

**Question - 36:**

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

**Ans:**

An Action Class is an adapter between the contents of an incoming HTTP rest and the corresponding business logic that should be executed to process this rest.

[View All Answers](#)

**Question - 37:**

What are the most common methods of Hibernate configuration?

**Ans:**

The most common methods of Hibernate configuration are:

- \* Programmatic configuration
- \* XML configuration (hibernate.cfg.xml)

[View All Answers](#)

**Question - 38:**

What is HQL?

**Ans:**

HQL stands for Hibernate Query Language. Hibernate allows the user to express queries in its own portable SQL extension and this is called as HQL. It also allows the user to express in native SQL.

[View All Answers](#)

**Question - 39:**



What is object/relational mapping metadata?

**Ans:**

ORM tools require a metadata format for the application to specify the mapping between classes and tables, properties and columns, associations and foreign keys, Java types and SQL types. This information is called the object/relational mapping metadata. It defines the transformation between the different data type systems and relationship representations.

[View All Answers](#)

**Question - 40:**

What are POJOs?

**Ans:**

POJO stands for plain old java objects. These are just basic JavaBeans that have defined setter and getter methods for all the properties that are there in that bean. Besides they can also have some business logic related to that property. Hibernate applications works efficiently with POJOs rather than simple java classes.

[View All Answers](#)

**Question - 41:**

What should SessionFactory be placed so that it can be easily accessed?

**Ans:**

As far as it is compared to J2EE environment, if the SessionFactory is placed in JNDI then it can be easily accessed and shared between different threads and various components that are hibernate aware. You can set the SessionFactory to a JNDI by configuring a property hibernate.session\_factory\_name in the hibernate.properties file.

[View All Answers](#)

**Question - 42:**

What are Callback interfaces?

**Ans:**

These interfaces are used in the application to receive a notification when some object events occur. Like when an object is loaded, saved or deleted. There is no need to implement callbacks in hibernate applications, but they're useful for implementing certain kinds of generic functionality.

[View All Answers](#)

**Question - 43:**

What the Core interfaces are of hibernate framework?

**Ans:**

There are many benefits from these. Out of which the following are the most important one.

Session Interface : This is the primary interface used by hibernate applications. The instances of this interface are lightweight and are inexpensive to create and destroy. Hibernate sessions are not thread safe.

SessionFactory Interface : This is a factory that delivers the session objects to hibernate application. Generally there will be a single SessionFactory for the whole application and it will be shared among all the application threads.

Configuration Interface : This interface is used to configure and bootstrap hibernate. The instance of this interface is used by the application in order to specify the location of hibernate specific mapping documents.

Transaction Interface : This is an optional interface but the above three interfaces are mandatory in each and every application. This interface abstracts the code from any kind of transaction implementations such as JDBC transaction, JTA transaction.

Query and Criteria Interface : This interface allows the user to perform queries and also control the flow of the query execution.

[View All Answers](#)

**Question - 44:**

What are the benefits of ORM and Hibernate?

**Ans:**

There are many benefits from these. Out of which the following are the most important one.

Productivity : Hibernate reduces the burden of developer by providing much of the functionality and let the developer to concentrate on business logic.

Maintainability :As hibernate provides most of the functionality, the LOC for the application will be reduced and it is easy to maintain. By automated object/relational persistence it even reduces the LOC.

Performance : Hand-coded persistence provided greater performance than automated one. But this is not true all the times. But in hibernate, it provides more optimization that works all the time there by increasing the performance. If it is automated persistence then it still increases the performance.

Vendor independence : Irrespective of the different types of databases that are there, hibernate provides a much easier way to develop a cross platform application.

[View All Answers](#)

**Question - 45:**

What is meant by full object mapping?

**Ans:**

Full object mapping supports sophisticated object modeling: composition, inheritance, polymorphism and persistence. The persistence layer implements transparent persistence; persistent classes do not inherit any special base class or have to implement a special interface. Efficient fetching strategies and caching strategies are implemented transparently to the application.

[View All Answers](#)

**Question - 46:**

What is a meant by medium object mapping?

**Ans:**

The application is designed around an object model. The SQL code is generated at build time. And the associations between objects are supported by the persistence mechanism, and queries are specified using an object-oriented expression language. This is best suited for medium-sized applications with some complex transactions. Used when the mapping exceeds 25 different database products at a time.

[View All Answers](#)

**Question - 47:**

What is meant by light object mapping?

**Ans:**

The entities are represented as classes that are mapped manually to the relational tables. The code is hidden from the business logic using specific design patterns. This approach is successful for applications with a less number of entities, or applications with common, metadata-driven data models. This approach is most known to all.

[View All Answers](#)

**Question - 48:**

Why do you need ORM tools like hibernate?

**Ans:**

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

[View All Answers](#)

**Question - 49:**

What is a pure relational ORM?

**Ans:**

The entire application, including the user interface, is designed around the relational model and SQL-based relational operations.

[View All Answers](#)

**Question - 50:**

What are the different levels of ORM quality?

**Ans:**

There are four levels defined for ORM quality.

Pure relational  
object mapping  
Medium object mapping  
Full object mapping

[View All Answers](#)

**Question - 51:**

What does an ORM solution comprises of?

**Ans:**

It should have an API for performing basic CRUD (Create, Read, Update, Delete) operations on objects of persistent classes. Should have a language or an API for specifying queries that refer to the classes and the properties of classes. An ability for specifying mapping metadata. It should have a technique for ORM implementation to interact with transactional objects to perform dirty checking, lazy association fetching, and other optimization functions.

[View All Answers](#)

**Question - 52:**

What is ORM?

**Ans:**

ORM stands for Object/Relational mapping. It is the programmed and translucent perseverance of objects in a Java application in to the tables of a relational database using the metadata that describes the mapping between the objects and the database. It works by transforming the data from one representation to another.

[View All Answers](#)

**Question - 53:**

What is Hibernate?

**Ans:**

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.

[View All Answers](#)

Interview Questions Answers.ORG

# Java Programing Most Popular & Related Interview Guides

- 1 : [Core Java Interview Questions and Answers.](#)
- 2 : [IBM WebSphere Interview Questions and Answers.](#)
- 3 : [Advanced Java Interview Questions and Answers.](#)
- 4 : [Spring Framework Interview Questions and Answers.](#)
- 5 : [Full Stack Developer \(Java\) Interview Questions and Answers.](#)
- 6 : [JSF Interview Questions and Answers.](#)
- 7 : [JDBC Interview Questions and Answers.](#)
- 8 : [Java Swing Programming Interview Questions and Answers.](#)
- 9 : [Java JSP Programming Interview Questions and Answers.](#)
- 10 : [J2EE Interview Questions and Answers.](#)

Follow us on FaceBook

[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/  
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)