



## Top 50 Hibernate Interview Questions

### 1. What's Hibernate?

Hibernate is a popular framework of Java which allows an efficient Object Relational mapping using configuration files in XML format. After java objects mapping to database tables, database is used and handled using Java objects without writing complex database queries.

### 2. What is ORM?

ORM (Object Relational Mapping) is the fundamental concept of Hibernate framework which maps database tables with Java Objects and then provides various API's to perform different types of operations on the data tables.

### 3. How properties of a class are mapped to the columns of a database table in Hibernate?

Mappings between class properties and table columns are specified in XML file as in the below example:

### 4. What's the usage of Configuration Interface in hibernate?

Configuration interface of hibernate framework is used to configure hibernate. It's also used to bootstrap hibernate. Mapping documents of hibernate are located using this interface.

### 5. How can we use new custom interfaces to enhance functionality of built-in interfaces of hibernate?

We can use extension interfaces in order to add any required functionality which isn't supported by built-in interfaces.

### 6. Should all the mapping files of hibernate have .hbm.xml extension to work properly?

No, having .hbm.xml extension is a convention and not a requirement for hibernate mapping file names. We can have any extension for these mapping files.

### 7. How do we create session factory in hibernate?

-  
To create a session factory in hibernate, an object of configuration is created first which refers to the path of configuration file and then for that configuration, session factory is created as given in the example below:

```
[crayon-582f0a988183a736575923/]
```

### **8. What are POJOs and what's their significance?**

POJOs( Plain Old Java Objects) are java beans with proper getter and setter methods for each and every properties.

Use of POJOs instead of simple java classes results in an efficient and well constructed code.

### **9. What's HQL?**

HQL is the query language used in Hibernate which is an extension of SQL. HQL is very efficient, simple and flexible query language to do various type of operations on relational database without writing complex database queries.

### **10. How can we invoke stored procedures in hibernate?**

In hibernate we can execute stored procedures using code as below:

```
[xml]
```

```
{ ? = call selectStudents() }
```

```
[/xml]
```

### **11. What is criteria API?**

Criteria is a simple yet powerful API of hibernate which is used to retrieve entities through criteria object composition.

### **12. What are the benefits of using Hibernate template?**

Following are some key benefits of using Hibernate template:

- a. Session closing is automated.
- b. Interaction with hibernate session is simplified.
- c. Exception handling is automated.

### **13. How can we see hibernate generated SQL on console?**

We need to add following in hibernate configuration file to enable viewing SQL on the console for debugging purposes:

```
[xml]
```

```
true
```

[/xml]

#### **14. What are the two types of collections in hibernate?**

Following are the two types of collections in hibernate:

- a. Sorted Collection
- b. Order Collection

#### **15. What's the difference between session.save() and session.saveOrUpdate() methods in hibernate?**

Session.save() method saves a record only if it's unique with respect to its primary key and will fail to insert if primary key already exists in the table.

saveOrUpdate() method inserts a new record if primary key is unique and will update an existing record if primary key exists in the table already.

#### **16. What the benefits are of hibernate over JDBC?**

- a. Hibernate can be used seamlessly with any type of database as its database independent while in case of JDBC, developer has to write database specific queries.
- b. Using hibernate, developer doesn't need to be an expert of writing complex queries as HQL simplifies query writing process while in case of JDBC, its job of developer to write and tune queries.
- c. In case of hibernate, there is no need to create connection pools as hibernate does all connection handling automatically while in case of JDBC, connection pools need to be created.

#### **17. How can we get hibernate statistics?**

We can get hibernate statistics using getStatistics() method of SessionFactory class as shown below:

```
SessionFactory.getStatistics()
```

#### **18. What is transient instance state in Hibernate?**

If an instance is not associated with any persistent context and also, it has never been associated with any persistent context, then it's said to be in transient state.

#### **19. How can we reduce database write action times in Hibernate?**

Hibernate provides dirty checking feature which can be used to reduce database write times. Dirty checking feature of hibernate updates only those fields which require a change while keeps others unchanged.

#### **20. What's the usage of callback interfaces in hibernate?**

Callback interfaces of hibernate are useful in receiving event notifications from objects. For example, when an object is loaded or deleted, an event is generated and notification is sent using callback interfaces.

#### **21. When an instance goes in detached state in hibernate?**

When an instance was earlier associated with some persistent context (e.g. a table) and is no longer associated, it's called to be in detached state.

**22. What the four ORM levels are in hibernate?**

Following are the four ORM levels in hibernate:

- a. Pure Relational
- b. Light Object Mapping
- c. Medium Object Mapping
- d. Full Object Mapping

**23. What's transaction management in hibernate? How it works?**

Transaction management is the process of managing a set of statements or commands. In hibernate; transaction management is done by transaction interface as shown in below code:

```
[java]
Session s = null;
Transaction tr = null;
try {
    s = sessionFactory.openSession();
    tr = s.beginTransaction();
    doTheAction(s);
    tr.commit();
} catch (RuntimeException exc) {
    tr.rollback();
} finally {
    s.close();
}
```

```
[/java]
```

**24. What the two methods are of hibernate configuration?**

We can use any of the following two methods of hibernate configuration:

- a. XML based configuration ( using hibernate.cfg.xml file)
- b. Programmatic configuration ( Using code logic)

**25. What is the default cache service of hibernate?**

Hibernate supports multiple cache services like EHCache, OSCache, SWARMCache and TreeCache and default cache service of hibernate is EHCache.

**26. What are the two mapping associations used in hibernate?**

In hibernate; we have following two types of mapping associations between entities:

- a. One-to-One Association
- b. Many-to-Many Association

**27. What's the usage of Hibernate QBC API?**

Hibernate Query By Criteria (QBC) API is used to create queries by manipulation of criteria objects at runtime.

**28. In how many ways, objects can be fetched from database in hibernate?**

Hibernate provides following four ways to fetch objects from database:

- a. Using HQL
- b. Using identifier
- c. Using Criteria API
- d. Using Standard SQL

### **29. How primary key is created by using hibernate?**

Database primary key is specified in the configuration file hbm.xml. Generator can also be used to specify how primary key is being created in the database.

In the below example, deptId acts as primary key:

[xml]

[/xml]

### **30. How can we reattach any detached objects in Hibernate?**

Objects which have been detached and are no longer associated with any persistent entities can be reattached by calling session.merge() method of session class.

### **31. What are different ways to disable hibernate second level cache?**

Hibernate second level cache can be disabled using any of the following ways:

- a. By setting use\_second\_level\_cache as false.
- b. By using CACHEMODE.IGNORE
- c. Using cache provider as org.hibernate.cache.NoCacheProvider

### **32. What is ORM metadata?**

All the mapping between classes and tables, properties and columns, Java types and SQL types etc is defined in ORM metadata.

### **33. Which one is the default transaction factory in hibernate?**

With hibernate 3.2, default transaction factory is JDBCTransactionFactory.

### **34. What's the role of JMX in hibernate?**

Java Applications and components are managed in hibernate by a standard API called JMX API. JMX provides tools for development of efficient and robust distributed, web based solutions.

### **35. How can we bind hibernate session factory to JNDI ?**

Hibernate session factory can be bound to JNDI by making configuration changes in hibernate.cfg file.

### **36. In how many ways objects can be identified in Hibernate?**

Object identification can be done in hibernate in following three ways:

- a. Using Object Identity: Using == operator.

- b. Using Object Equality: Using equals() method.
- c. Using database identity: Relational database objects can be identified if they represent same row.

### **37. What different fetching strategies are of hibernate?**

Following fetching strategies are available in hibernate:

1. Join Fetching
2. Batch Fetching
3. Select Fetching
4. Sub-select Fetching

### **38. How mapping of java objects is done with database tables?**

To map java objects with database tables, we need to have Java beans properties names same as column names of a database table. Then mapping is provided in hbm.xml file as given below:

[xml]

[/xml]

### **39. What are derived properties in hibernate?**

Derived properties are those properties which are not mapped to any columns of a database table. Such properties are calculated at runtime by evaluation of any expressions.

### **40. What is meant by a Named SQL Query in hibernate and how it's used?**

Named SQL queries are those queries which are defined in mapping file and are called as required anywhere.

For example, we can write a SQL query in our XML mapping file as follows:

[xml]

```
SELECT std.STUDENT_ID AS {std.STUDENT_ID},  
std.STUDENT_DISCIPLINE AS {std.discipline},
```

```
FROM Student std WHERE std.NAME LIKE :name
```

[/xml]

Then this query can be called as follows:

[java]

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

```
[/java]
```

#### **41. What's the difference between load() and get() method in hibernate?**

Load() methods results in an exception if the required records isn't found in the database while get() method returns null when records against the id isn't found in the database.

So, ideally we should use Load() method only when we are sure about existence of records against an id.

#### **42. What's the use of version property in hibernate?**

Version property is used in hibernate to know whether an object is in transient state or in detached state.

#### **43. What is attribute oriented programming?**

In Attribute oriented programming, a developer can add Meta data (attributes) in the java source code to add more significance in the code. For Java (hibernate), attribute oriented programming is enabled by an engine called XDoclet.

#### **44. What's the use of session.lock() in hibernate?**

session.lock() method of session class is used to reattach an object which has been detached earlier. This method of reattaching doesn't check for any data synchronization in database while reattaching the object and hence may lead to lack of synchronization in data.

#### **45. Does hibernate support polymorphism?**

Yes, hibernate fully supports polymorphism. Polymorphism queries and polymorphism associations are supported in all mapping strategies of hibernate.

#### **46. What the three inheritance models are of hibernate?**

Hibernate has following three inheritance models:

- a. Tables Per Concrete Class
- b. Table per class hierarchy
- c. Table per sub-class

#### **47. How can we map the classes as immutable?**

If we don't want an application to update or delete objects of a class in hibernate, we can make the class as immutable by setting mutable=false

#### **48. What's general hibernate flow using RDBMS?**

General hibernate flow involving RDBMS is as follows:

- a. Load configuration file and create object of configuration class.
- b. Using configuration object, create sessionFactory object.
- c. From sessionFactory, get one session.

- d. Create HQL query.
- e. Execute HQL query and get the results. Results will be in the form of a list.

#### 49. What is Light Object Mapping?

Light Object Mapping is one of the levels of ORM quality in which all entities are represented as classes and they are mapped manually.

#### 50. What's difference between managed associations and hibernate associations?

Managed associations relate to container management persistence and are bi-directional while hibernate associations are unidirectional.

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

Java	MIS	MongoDB	BigData	Cassandra
Web Services	SQLite	JSP	Informatica	Accounting
SAP Training	Python	Excel	ASP Net	HBase
Project Management	Test Management	Business Analyst	Ethical Hacking	PMP
Live Project	SoapUI	Photoshop	Manual Testing	Mobile Testing
Selenium	CCNA	AngularJS	NodeJS	PLSQL

Stay updated with new  
courses at Guru99  
Join our Newsletter