Top 50 Hibernate Interview Questions That Are A Must

Last updated on Jul 20,2020 *26.9K Views*

Vaishnavi M R

Hibernate is one of the most widely used ORM tools for building Java applications. It is used in enterprise applications for database operations. So, this article on hibernate interview questions will help you to brush up your knowledge before the interview.

If you are a fresher or an experienced, this is the right platform for you which will help you to start your preparation for the Hibernate job roles.

Let's begin by taking a look at the most frequently asked questions in Hibernate Interview Questions.

For better understanding, I have divided the rest of the Hibernate Framework Interview Questions into the following sections:

Let's begin!

Hibernate Interview Questions for beginners

Q1. What is Hibernate?

Hibernate is one of the most popular *Java frameworks* that simplify the development of Java application to interact with the database. It is an Object-relational mapping (ORM) tool. Hibernate also provides a reference implementation of Java API.

It is referred as a framework which comes with an abstraction layer and also handles the implementations internally. The implementations include tasks like writing a query for *CRUD* operations or establishing a connection with the databases, etc.

Hibernate develops persistence logic, which stores and processes the data for longer use. It is a lightweight tool and most importantly **open-sourced** which gives it an edge over other frameworks.

Q2. What are the major advantages of Hibernate Framework?

- It is open-sourced and lightweight.
- Performance of Hibernate is very fast.
- Helps in generating database independant queries.
- Provides facilities to automatically create a table.
- It provides query statistics and database status.

Q3. What are the advantages of using Hibernate over JDBC?

Major advantages of using Hibernate over JDBC are:

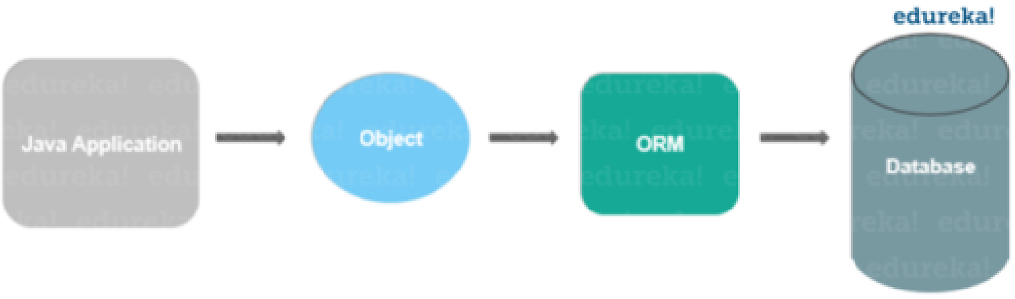
1. Hibernate eliminates a lot of boiler-plate code that comes with *JDBC API*, the code looks cleaner and readable.
2. This Java framework supports *inheritance*, associations, and collections. These features are actually not present in JDBC.

management using *commit* and *rollback*.

5. JDBC throws SQLException that is a checked exception, so you have to write a lot of try-catch block code. Hibernate wraps JDBC exceptions and throw *JDBCException* or *HibernateException* which are the unchecked exceptions, so you don't have to write code to handle it has built-in transaction management which helps in removing the usage of try-catch blocks.

Q4. What is an ORM tool?

It is basically a technique that maps the object that is stored in the database. An ORM tool helps in simplifying data creation, manipulation, and access. It internally uses the Java API to interact with the databases.



Q5. Why use Hibernate Framework?

Hibernate overcomes the shortcomings of other technologies like *JDBC*.

- It overcomes the database dependency faced in the JDBC.
- Changing of the databases cost a lot working on JDBC, hibernate overcomes this problem with flying colors.
- Code portability is not an option while working on JDBC. This is easily handled by Hibernate.
- Hibernate strengthens the object level relationship.
- It overcomes the *exception-handling* part which is mandatory while working on JDBC.
- It reduces the length of code with increased readability by overcoming the boilerplate problem.

Q6. What are the different functionalities supported by Hibernate?

- Hibernate is an ORM tool.
- Hibernate uses Hibernate Query Language(HQL) which makes it database-independent.
- It supports auto DDL operations.
- This Java framework also has an *Auto Primary Key Generation* support.
- Supports cache memory.
- Exception handling is not mandatory in the case of Hibernate.

Q7. What are the technologies that are supported by Hibernate?

Hibernate supports a variety of technologies, like:

- XDoclet Spring
- *Maven*
- Eclipse Plug-ins
- *J2EE*

Q8. What is HQL?

HQL is the acronym of Hibernate Query Language. It is an Object-Oriented Query Language and is independent of the database.

Q9. How to achieve mapping in Hibernate?

Association mappings are one of the key features of Hibernate. It supports the same associations as the relational database model. They are:

- One-to-One associations
- Many-to-One associations
- Many-to-Many associations

You can map each of them as a uni- or bidirectional association.

Q10. Name some of the important interfaces of Hibernate framework?

Hibernate interfaces are:

- **SessionFactory** (org.hibernate.SessionFactory)
- **Session** (org.hibernate.Session)
- **Transaction** (org.hibernate.Transaction)

Q11. What is One-to-One association in Hibernate?

Q12. What is One-to-Many association in Hibernate?

In this type of association, one object can be associated with multiple/different objects. Talking about the mapping, the One-to-Many mapping is implemented using a *Set Java* collection that does not have any redundant element. This *One-to-Many* element of the set indicates the relation of one object to multiple objects.

Q13. What is Many-to-Many association in Hibernate?

Many-to-Many mapping requires an entity attribute and a *@ManyToMany* annotation. It can either be unidirectional and bidirectional. In **Unidirectional**, the attributes model the association and you can use it to navigate it in your domain model or JPQL queries. The annotation tells Hibernate to map a Many-to-Many association. The **bidirectional** relationship, mapping allows you to navigate the association in both directions.

Q14. How to integrate Hibernate and Spring?

Spring is also one of the most commonly used Java frameworks in the market today. Spring is a JavaEE Framework and Hibernate is the most popular ORM framework. This is why Spring Hibernate combination is used in a lot of enterprise applications.

Following are the steps you should follow to integrate Spring and Hibernate.

1. Add Hibernate-entity manager, Hibernate-core and Spring-ORM dependencies.
2. Create Model classes and corresponding DAO implementations for database operations. The DAO classes will use *SessionFactory* that will be injected by the Spring Bean configuration.
3. Note that you don't need to use Hibernate Transaction Management, as you can leave it to the Spring declarative transaction management using *@Transactional annotation.*

Q15. What do you mean by Hibernate Configuration File?

Hibernate Configuration File mainly contains database-specific configurations and are used to initialize *SessionFactory*. Some important parts of the Hibernate Configuration File are Dialect information, so that hibernate knows the database type and mapping file or class details.

Hibernate Interview Questions for intermediate

Q16. Mention some important annotations used for Hibernate mapping?

Hibernate supports JPA annotations. Some of the major annotations are:

1. **javax.persistence.Entity:** This is used with model classes to specify they are entity beans.
2. **javax.persistence.Table:** It is used with entity beans to define the corresponding table name in the database.
3. **javax.persistence.Access:** Used to define the access type, field or property. The default value is field and if you want Hibernate to use the getter/setter methods then you need to set it to a property.
4. **javax.persistence.Id:** Defines the primary key in the entity bean.
5. **javax.persistence.EmbeddedId:** It defines a composite primary key in the entity bean.
6. **javax.persistence.Column:** Helps in defining the column name in the database table.
7. **javax.persistence.GeneratedValue:** It defines the strategy to be used for the generation of the primary key. It is also used in conjunction with *javax.persistence.GenerationType* enum.

Q17. What is Session in Hibernate and how to get it?

Hibernate Session is the interface between Java application layer and Hibernate. It is used to get a physical connection with the database. The *Session* object created is lightweight and designed to be instantiated each time an interaction is needed with the database. This *Session* provides methods to create, read, update and delete operations for a constant object. To get the Session, you can execute HQL queries, SQL native queries using the *Session* object.

Q18. What is Hibernate SessionFactory?

SessionFactory is the factory class that is used to get the Session objects. The SessionFactory is a heavyweight object so usually, it is created during application startup and kept for later use. This *SessionFactory* is a thread-safe object which is used by all the threads of an application. If you are using multiple databases then you would have to create multiple *SessionFactory* objects.

Q19. What is the difference between openSession and getCurrentSession?

This *getCurrentSession()* method returns the session bound to the context and for this to work, you need to configure it in Hibernate configuration file. Since this session object belongs to the context of Hibernate, it is okay if you don't close it. Once the *SessionFactory* is closed, this session object gets closed.

openSession() method helps in opening a new session. You should close this session object once you are done with all the database operations. And also, you should open a new session for each request in a multi-threaded environment.

Q20. What do you mean by Hibernate configuration file?

The following steps help in configuring Hibernate file:

1. First, identify the POJOs (Plain Old Java Objects) that have a database representation.
2. Identify which properties of POJOs need to be continued.
3. Annotate each of the POJOs in order to map the Java objects to columns in a database table.
4. Create a database schema using the schema export tool which uses an existing database, or you can create your own database schema.
5. Add Hibernate Java libraries to the application's classpath.
6. Create a Hibernate *XML configuration file* that points to the database and the mapped classes.
7. In the Java application, you can create a Hibernate Configuration object that refers to your XML configuration file.
8. Also, build a Hibernate SessionFactory object from the Configuration object.
9. Retrieve the Hibernate Session objects from the SessionFactory and write down the data access logic for your application (create, retrieve, update, and delete).

Q21. What are the key components of a Hibernate configuration object?

The configuration provides 2 key components, namely:

- *Database Connection:* This is handled by one or more configuration files.
- *Class Mapping setup*: It helps in creating the connection between Java classes and database tables.

Q22. Discuss the Collections in Hibernate

Hibernate provides the facility to persist the Collections. A *Collection* basically can be a List, Set, Map, Collection, Sorted Set, Sorted Map. java.util.List, java.util.Set, java.util.Collection, etc, are some of the real interface types to declared the persistent collection-value fields. Hibernate injects persistent Collections based on the type of interface. The collection instances generally behave like the types of value behavior.

Q23. What are the collection types in Hibernate?

There are five collection types in hibernate used for one-to-many relationship mappings.

- Bag
- Set
- List
- Array
- Map

Q24. What is a Hibernate Template class?

When you integrate Spring and Hibernate, Spring ORM provides two helper classes – HibernateDaoSupport and HibernateTemplate. The main reason to use them was to get two things, the Session from Hibernate and Spring Transaction Management. However, from Hibernate 3.0.1, you can use the SessionFactory getCurrentSession() method to get the current session. The major advantage of using this Template class is the **exception translation** but that can be achieved easily by using *@Repository* annotation with service classes.

Q25. What are the benefits of using Hibernate template?

The following are the benefits of using this Hibernate template class:

- Automated Session closing ability.
- The interaction with the Hibernate Session is simplified.
- Exception handling is automated.

Q26. Which are the design patterns that are used in Hibernate framework?

- *Data Mapper:* A layer of the map that moves data between objects and a database while keeping it independent of each other and the map itself.
- *Proxy Pattern:* It is used for lazy loading.
- *Factory Pattern:* Used in SessionFactory.

Q27. Define Hibernate Validator Framework

Data validation is considered as an integral part of any application. Also, data validation is used in the presentation layer with the use of Javascript and the server-side code before processing. It occurs before persisting it in order to make sure it follows the correct format. Validation is a cross-cutting task, so we should try to keep it apart from the business logic. This Hibernate Validator provides the reference implementation of bean validation specs.

Q28. What is Dirty Checking in Hibernate?

Hibernate incorporates *Dirty Checking* feature that permits developers and users to avoid time-consuming write actions. This Dirty Checking feature changes or updates fields that need to be changed or updated, while keeping the remaining fields untouched and unchanged.

Q29. How can you share your views on mapping description files?

- Mapping description files are used by the Hibernate to configure functions.
- These files have the ***.hbm** extension, which facilitates the mapping between database tables and Java class.
- Whether to use mapping description files or not this entirely depends on business entities.

Q30. What is meant by Light Object Mapping?

The means that the syntax is hidden from the business logic using specific design patterns. This is one of the valuable levels of ORM quality and this Light Object Mapping approach can be successful in case of applications where there are very fewer entities, or for applications having data models that are metadata-driven.

Hibernate Interview Questions for experienced

Q31. What is meant by Hibernate tuning?

Optimizing the performance of Hibernate applications is known as Hibernate tuning.

The performance tuning strategies for Hibernate are:

1. SQL Optimization
2. Session Management
3. Data Caching

Q32. What is Transaction Management in Hibernate? How does it work?

Transaction Management is a property which is present in the Spring framework. Now, what role does it play in Hibernate?

Transaction Management is a process of managing a set of commands or statements. In hibernate, Transaction Management is done by transaction interface. It maintains abstraction from the transaction implementation (JTA, JDBC). A transaction is associated with Session and is instantiated by calling *session.beginTransaction()*.

Q33. How do you integrate Hibernate with Struts2 or Servlet web applications?

You can integrate any Struts application with Hibernate. There are no extra efforts required.

1. Register a custom *ServletContextListener*.
2. In the *ServletContextListener* class, first, initialize the Hibernate Session, store it in the servlet context.
3. Action class helps in getting the Hibernate Session from the servlet context, and perform other Hibernate task as normal.

Q34. What are the different states of a persistent entity?

It may exist in one of the following 3 states:

- Transient: This is not associated with the Session and has no representation in the database.
- Persistent: You can make a transient instance persistent by associating it with a Session.
- Detached: If you close the Hibernate Session, the persistent instance will become a detached instance.

Q35. How can the primary key be created by using Hibernate?

A Primary key is a special relational database table column designated to uniquely identify all table records. It is specified in the configuration file *hbm.xml*. The generator can also be used to specify how a Primary key can be created in the database.

Q36. Explain about Hibernate Proxy and how it helps in Lazy loading?

- Hibernate uses a proxy object in order to support *Lazy loading*.
- When you try loading data from tables, Hibernate doesn't load all the mapped objects.
- After you reference a child object through getter methods, if the linked entity is not present in the session cache, then the proxy code will be entered to the database and load the linked object.
- It uses Java assist to effectively and dynamically generate sub-classed implementations of your entity objects.

Q37. How can we see Hibernate generated SQL on console?

In order to view the SQL on a console, you need to add following in Hibernate configuration file to enable viewing SQL on the console for debugging purposes:

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

Q38. What is Query Cache in Hibernate?

Hibernate implements a separate cache region for queries resultset that integrates with the Hibernate second-level cache. This is also an optional feature and requires a few more steps in code.

***Note:** This is only useful for queries that are run frequently with the same parameters.*

Q39. What is the benefit of Native SQL query support in Hibernate?

Hibernate provides an option to execute Native SQL queries through the use of the *SQLQuery* object. For normal scenarios, it is however not the recommended approach because you might lose other benefits like Association and Hibernate first-level caching.

Native SQL Query comes handy when you want to execute database-specific queries that are not supported by Hibernate API such query hints or the *Connect* keyword in Oracle Database.

Q40. What is Named SQL Query?

Hibernate provides another important feature called Named Query using which you can define at a central location and use them anywhere in the code.

You can create named queries for both HQL as well as for Native SQL. These Named Queries can be defined in Hibernate mapping files with the help of JPA annotations @NamedQuery and @NamedNativeQuery.

Q41. When do you use merge() and update() in Hibernate?

This is one of the tricky Hibernate Interview Questions asked.

update(): If you are sure that the Hibernate Session does not contain an already persistent instance with the same *id* .
merge(): Helps in merging your modifications at any time without considering the state of the Session.

Q42. Difference between get() vs load() method in Hibernate?