



Hibernate Interview Questions and Answers

Introduction

Hibernate is a crucial skill for Java programmers who want to optimize database interactions using Object-Relational Mapping (ORM). This tutorial will walk you through essential concepts such as Session Management, Caching, and HQL to make sure you can develop high-performance data-driven applications. Since hibernate is a commonly used ORM solution, you can expect many questions about it in tech interviews. Below is a list of the top 20 **Hibernate interview questions and answers** for both freshers and experienced professionals.

Are you ready to take your skills to the next level and master persistent data? Take your skills to the next level by enrolling in our [Hibernate Certification Course in Chennai](#) today.

List of Hibernate Interview Questions for Freshers

1. What is hibernate framework?

2. Explain the ORM tool.
3. What in Hibernate is a session?
4. Regarding the claim that “a session is a thread-safe object,” what are your thoughts?
5. Could you elaborate on what sluggish loading in hibernation means?
6. Could you elaborate on what sluggish loading in hibernation means?
7. Specify the requirements using Hibernate.
8. Benefits of Hibernate usage?
9. What does “lightweight” mean in the Hibernate framework?
10. Describe “persistence.”

Explore our complete [Hibernate course syllabus](#).

Hibernate Interview Questions and Answers for Freshers

1. What is hibernate framework?

An open-source framework called Hibernate facilitates relational database information storage. Using ORM, it effortlessly translates Java objects to matching database tables. CRUD (Create, Read, Update, Delete) actions automate database activities.

Consequently, Hibernate offers a smooth process and improved interactive characteristics, negating the necessity to write SQL queries.

2. Explain the ORM tool.

An Object Relational Mapping (ORM) tool uses an internal Java API to communicate with the databases, making data creation, management, and access easier. It is a method for mapping database-stored items.

3. What in Hibernate is a session?

An object called a session keeps the Java application and database connected. Additionally, a session contains methods like load(), get(), update(), delete(), and persist() that can be used to store, retrieve, alter, or remove data from a database. It also contains factory methods to return objects like queries, criteria, and transactions.

4. Regarding the claim that “a session is a thread-safe object,” what are your thoughts?

No, a session is not a thread-safe object, meaning many threads may access its contents simultaneously.

5. Could you elaborate on what sluggish loading in hibernation means?

Lazy loading primarily enhances application speed by assisting in the on-demand loading of child objects. Note that this feature has been enabled by default since Hibernate version 3. This means that child objects are not loaded until the parent object is loaded.

**Take your Knowledge
Test Report**

Check your Score



6. Could you elaborate on what sluggish loading in hibernation means?

The following are the main Hibernate framework interfaces: Criteria, Transaction, Configuration, SessionFactory, Session, and Query.

7. Specify the requirements using Hibernate.

Object-oriented criteria queries are created and executed using the objects of the criteria.

8. Benefits of Hibernate usage?

- Hibernate ensures that database changes are applied completely, preventing data corruption.
- Similar to SQL, Hibernate has its own query language, called HQL.
- Based on the Java class structure, HQL creates automatic database tables and accelerates performance.

9. What does “lightweight” mean in the Hibernate framework?

An application, computer program, or gadget that uses little system resources because of its small memory footprint (RAM) and low CPU utilization is referred to as “lightweight” in the context of computers.

10. Describe “persistence.”

Persistence in the context of Java refers to information and entities that survive beyond the creation process.

Read this [Hibernate tutorial for beginners](#) for a comprehensive understanding of ORM.

List of Hibernate Interview Questions for Experienced

1. Define Java Persistence API (JPA).
2. Describe the idea underlying the mapping of hibernation inheritance.
3. What benefits does the Java Hibernate framework offer?
4. Is an SQL injection attack a possibility with Hibernate?
5. Describe the hibernate mapping file.
6. What does the Hibernate criteria API mean?
7. Explain many-to-many associations
8. What is the purpose of the hibernate “session.lock()” method?
9. What is hibernate caching?
10. Which Hibernate concurrency strategies are available?

Get to know the best [Hibernate salary for freshers](#) here.

Hibernate Technical Interview Questions and Answers for Experienced

1. Define Java Persistence API (JPA).

A group of classes and methods known as the Java Persistence API are used to use ORM to persist or store enormous amounts of data in databases. The JPA Persistence Framework must be adhered to:

- The amount of boilerplate code required for typical database operations like GET, PUT, POST, etc. has decreased with **Spring Data JPA**.
- **Spring Repository**: This is an addition to Spring Repository that offers APIs for pagination, sorting, and basic CRUD activities.

2. Describe the idea underlying the mapping of hibernation inheritance.

Relational databases can be mapped to object-oriented inheritance relationships using Hibernate's inheritance mapping feature. In programming, it also guarantees the organization of class hierarchies while storing data in database tables. Hibernate uses the tables below to handle the mapping.

- Single Table
- Table Per Class
- Joined Table Inheritance

3. What benefits does the Java Hibernate framework offer?

The many benefits of the Hibernate framework are listed below:

- It moves quickly, and it is free and open source.
- Cutting out boilerplate code shortens code and frees up developers to work on other projects.
- It makes the object-level relationship stronger and makes it easier to create standalone database queries.
- It offers resources for automatically constructing tables.
- Integrating it with other Java Enterprise Edition (EE) frameworks is simple.

4. Is an SQL injection attack a possibility with Hibernate?

A critical weakness in online security is the SQL injection attack, which allows an attacker to tamper with database queries made by an application or website, giving the attacker access to view sensitive data that is typically irretrievable.

Additionally, it may allow an attacker to alter or remove data, harming the functionality of the program. SQL injection is not immune to Hibernate. However, by following recommended practices, SQL injection attacks can be avoided. It is usually a good idea to choose one of the choices listed below:

- Combine parameterized queries in prepared statements.
- Make use of stored procedures.
- Validate input to guarantee data sanity.

5. Describe the hibernate mapping file.

The entity bean fields and associated database column mappings are defined in an XML file called the Hibernate mapping file.

These files come in handy when the project needs to use third-party classes and hibernating cannot be used to use the JPA annotations.

In the preceding example, the mapping resource was defined in the configuration file as “SLAemp.hbm.xml.” Now let’s examine the sample hbm.xml file:

```
<?xml version = "1.0" encoding = "utf-8"?>

<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

  <!-- What class is mapped to what database table-->

  <class name = "SLAemp" table = "SLAemp">

    <meta attribute = "class-description">

      This class contains the details of employees of SLA.

    </meta>

    <id name = "id" type = "int" column = "employee_id">

      <generator class="native"/>

    </id>
```

```

    <property name = "fullName" column = "full_name" type = "string"/>

    <property name = "email" column = "email" type = "string"/>

</class>

</hibernate-mapping>

```

6. What does the Hibernate criteria API mean?

Hibernate's Criteria API facilitates the creation of dynamic criteria queries for the persistence database by developers. When developing dynamic queries, the Criteria API offers a more potent and adaptable option than HQL (Hibernate Query Language) queries.

Programmatically, development criteria query objects can be accessed with this API. The hibernating org. For these purposes, a criteria interface is employed.

The createCriteria() method in the Hibernate framework's Session interface takes the class or entity name of the persistent object input and returns an instance of the persistence object when the criteria query is run.

Additionally, it allows incorporating constraints to retrieve data from the database selectively, which is quite simple.

The add() method, which takes the org.hibernate.criterion, can be used to do it. A criterion item that stands for personal limitation.

```

javax.persistence.criteria.CriteriaBuilder

javax.persistence.criteria.CriteriaQuery

// Create CriteriaBuilder

CriteriaBuilder builder = session.getCriteriaBuilder();

// Create CriteriaQuery

CriteriaQuery<YourClass> criteria = builder.createQuery(YourClass.class);

```

Using the CriteriaQuery.where function, which is comparable to using the WHERE clause in a JPQL query, allows us to add limits to the CriteriaQuery.

7. Explain many-to-many associations.

A many-to-many association shows that two entities' instances are related in various ways. We might use the scenario where several students enroll in different classes and vice versa.

We formally describe this relationship by building a new table to house the foreign keys that are used to refer to each other by the course and student entities.

The `student_id` and `course_id` would constitute the composite primary key in this case, which is why the student-course table is referred to as the join table.

**Take your Knowledge
Test Report**

Check your Score



8. What is the purpose of the hibernate “`session.lock()`” method?

A disconnected object can be reattached to the session using the `session.lock()` method. Data synchronization between the database and the object in the persistence context may be lost as a result of this reattachment because of the `session.lock()` method does not check for it.

9. What is hibernate caching?

Hibernate caching is a technique for increasing application performance through the pooling of items in the cache to speed up query execution. Hibernate caching comes in very handy when you need to retrieve the same data repeatedly. We can retrieve the data directly from the cache without having to query the database. The application's throughput time is shortened as a result.

Hibernate Caching Types: First-Level Cache

- By default, this level is activated.

- The hibernate session object houses the first-level cache.
- The scope of the data contained here will not be exposed to the entire program because it is a part of the session object, even though an application may utilize more than one session object.

Cache at the Second Level:

- The SessionFactory object houses the second-level cache, which makes the data accessible to the entire program.
- By default, this is not available. It needs to be specifically activated.
- Cache providers include OS Cache, JBoss Cache, Swarm Cache, and EH (Easy Hibernate) Cache.

10. Which Hibernate concurrency strategies are available?

The mediators in charge of keeping and removing objects from the cache are concurrency strategies. The developer bears the obligation of specifying the appropriate technique for each persistent class and collection when enabling second-level caching. The concurrency strategies that are employed are as follows:

- **Transactional:** This is utilized when changing data, which almost certainly results in stale data. The application depends heavily on this prevention.
- **Read-Only:** This can only be used as reference data and is utilized when we don't want the data altered.
- **Read-Write:** In this scenario, data is mostly read and utilized when preventing stale data is crucial.
- **Non-strict Read-Write:** By employing this technique, consistency between the database and cache will be guaranteed. When stale data is not a major concern, and the data is modifiable, this technique can be applied.

Conclusion

Java professionals with Hibernate skills are in great demand. Your chances of succeeding in your upcoming Hibernate interview can be increased by being ready for these Hibernate interview questions and answers. Passing a Hibernate interview means you have to show proficiency in lazy loading, caching levels, and transaction management. Your ability to optimize database performance using effective ORM mapping is crucial for developing scalable Java applications.

Are you ready to become an IT professional? Unlock your career in our [software training institute in Chennai](#).