

[By OnlineInterviewQuestions.com](http://OnlineInterviewQuestions.com)

[Hibernate Interview Questions](#)

What is Hibernate?

Hibernate is an **ORM** (object-relational mapping) tool for Java programming language. It is developed by **Red Hat** and was started by Gavin King with his colleagues in 2001. The original aim was to offer better persistence capabilities than those offered by EJB2 (Enterprise Java Beans version 2), they wanted to simplify the complexities and supplement certain missing features.

Below we have provided **Hibernate Interview questions** that are specifically designed to give you the basic conceptual information at the beginning and later with the core nature of the subject. The information which has been provided will brush up your mind and will substantially contribute to impressing the interviewer so that you fly with high colors.

Quick Questions About Hibernate

Hibernate is written In	Java
Hibernate is a	object-relational mapping tool (ORM) for the Java programming language.
Hibernate is developed By	Red Hat Software
Hibernate Platform	Java Virtual Machine
Hibernate Licence	GNU Lesser General Public License
Hibernate Current Stable release	5.4.23

Q1. [What is Hibernate ?](#)

- **Hibernate** is a free software, which is distributed under GNU Lesser General public license 2.1. It is categorized under the Object Relational Mapping (ORM), which features the mapping of Java classes to data tables and mapping from Java data types to SQL data types.
- It is written in **Java** and is JVM (Java Virtual Machine) platform based.
- Hibernate provides the data query and retrieval facilities.
- It is one of the most widely used ORM tools for the Java applications.
- Hibernate is highly considered to be used in the enterprise applications for database operations.

Q2. [What is HQL in Hibernate?](#)

HQL is the acronym of Hibernate Query Language. It considers the java objects in a similar way as that of the SQL.

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

Q3. [Explain the advantages of Hibernate?](#)

Some of the advantages of Hibernate are:

- It provides Simple Querying of data.
- An application server is not required to operate.
- The complex associations of objects in the database can be manipulated.
- Database access is minimized with smart fetching strategies.
- It manages the mapping of Java classes to database tables without writing any code.
- Properties of XML file is changed in case of any required change in the database.

Q4. Why is ORM preferred over JDBC?

- It allows business code access the objects rather than Database tables.
- It hides the details of SQL queries from OO logic.
- It is based on JDBC “under hood”.
- Dealing with database implementation is not required.
- Entities are based on business concepts rather than database structures.
- It generates the automatic key and Transaction management.
- Application development is faster.

Q5. Mention the Key components of Hibernate?

The Key components of Hibernate are:

- **Session:** It is used to get a physical network with a database.
- **Transaction:** It represents the unit of work with a database.
- **Query:** It uses SQL and HQL string to retrieve the data from the database and create objects.
- **Criteria:** It is used to create and execute object-oriented queries and retrieve the objects.
- **Configuration:** It represents the properties of files required by Hibernate
- **Session Factory:** It configures hibernate for the application using the provided configuration file and instantiates the session object.

Q6. Explain Session object in Hibernate?

It is used to get a physical connection with a database. A session object is designed to instantiate each time an interaction is required with the database, whereas the persistent objects are retrieved using a session object.

The session objects are not thread-safe and must be created and destroyed as per the requirement.

Q7. Explain the Transaction object in Hibernate?

It represents a unit of work with the database and most of the RDBMS (Relational Database Management System) supports transaction functionality.

In Hibernate, **transactions** are managed by an underlying transaction manager and transaction from JDBC or JTA.

It is an optional object and the Hibernate Application do not use this interface, instead, they handle the

transactions in their code.

Q8. Explain the Criteria object in Hibernate?

Criteria objects are used to create and execute object-oriented Queries to retrieve the objects.

Q9. Explain the Query object in Hibernate?

These objects use SQL and HQL string to retrieve data from the database and create objects. An instance of Query is used to bind query parameters, restrict the number of results returned by the query and finally to execute the query.

Q10. Mention some of the databases that Hibernate supports?

Hibernate supports all the major **RDMS**. Following are the list of database engines supported by Hibernate:

- HSQL Database Engine
- DB2/NT
- Oracle
- Microsoft SQL Server Database
- Sybase SQL Server
- Informix Dynamic Server
- MySQL
- PostgreSQL
- FrontBase

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

It is similar to the **many-to-one association** and the difference lies in the column that will be set as a unique one. The many-to-one element is used to define one-to-one association.

To the defined variable a name attribute is set in the parent class and the column attribute is used to set column name in the parent table, which is unique so that only one object gets associated with another.

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

In this association, one object can be associated with multiple objects.

The One-to-Many mapping is implemented using a Set Java collection that does not have any redundant element.

A One-to-Many element of the set element indicates the relation of one object to multiple objects.

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

This association is the common type of association where one object can be associated with multiple objects. And Many-to-one element defines the Many-to-One association. To the defined variable, a name attribute is set in the parent class and column attribute sets the column name in the parent table.

Q14. What is Many-to-Many association in Hibernate?

The Many-to-Many element indicates the relation between one object to many other objects and column attribute is used to link intermediate columns. A Many-to-Many mapping is implemented using a Set Java collection that does not have any redundant element.

Q15. What is Hibernate caching?

Hibernate caches Query data and makes the application run faster.

If used correctly, the hibernate cache can be very useful in achieving the faster application running performance.

The main idea lying behind the cache is reducing the number of database queries, which results in reduced throughput time of the application.

Q16. What is first level cache in Hibernate?

It is session cache and mandatory cache. It is from first level cache through which all the requests must pass. The session object stores an object under its control before committing it to the database.

Q17. What is second level cache in Hibernate?

It is an optional cache. And, always the first level cache will be consulted before any attempt is performed to locate an object in the second level cache. This cache can be configured on a pre-collection and per-class basis and it is mainly responsible for caching objects across the sessions.

Q18. What is Query level cache in Hibernate?

In hibernate, a **cache query** can be implemented that results in sets and integrates closely with the second level cache. It is an optional feature and it requires two additional cache regions that can hold the cached query results and also the timestamps whenever a table is updated. This is useful only for the queries that run frequently holding the same parameters.

Q19. Explain Hibernate configuration file and Hibernate mapping file?

Hibernate configuration file:

It contains database specific configurations and is used to initialize SessionFactory.

It provides database credentials or JNDI resource information in the hibernate configuration XML file.

Dialect information is another important part of the hibernate configuration file.

Hibernate Mapping file:

It is used to define the database table column mappings and entity bean fields.

We use JPA annotations for mappings, but when we are using the third party classes sometimes XML mapping files becomes handy and we cannot use annotations.

Q20. What are concurrency strategies?

The concurrency strategies are the mediators who are responsible for storing items and retrieving them from the cache. In case of enabling a second level cache, the developer must decide for each persistent class and collection, and also which cache concurrency, has to be implemented.

Following are the concurrency strategies that can be implemented by the developer:

- **Transactional:** This strategy is used mostly to read data where the prevention of stale data is critical in concurrent transactions, in the unique case of an update.
- **Read- Only:** This strategy is compatible with the data that can't be modified. We can use it for reference data only.
- **Read-Write:** It is similar to transactional strategy. where we read mostly data and prevention of stale data is critical.
- **Non-strict-Read-Write:** This strategy assures no guarantee of consistency between the database and cache. We can use this strategy only if the data can be modified and a small likelihood of stale data is not the critical concern.

Q21. What is Lazy loading in hibernate ?

It is a technique in where the objects are loaded on the requirement basis. Since the Hibernate 3 version, the **lazy loading** is by default enabled so that the child objects are not loaded while the parent is loaded.

Q22. Explain the persistent classes in Hibernate?

In hibernate, the Java classes whose instances and objects are stored in database classes are called persistent classes.

Q23. Explain some of the elements of hbm.xml?

- It is used to define specific mappings from Java classes to database tables.
- It is used to define the mapping of unique ID attribute in class to the primary key of the database table.
- It is used to generate the primary key values automatically.
- It is used to map a Java class property to a column in the database table.
- It is used to map a java.util.set, java.util.Sortedset property in hibernate.
- It is used to map a java.util.List property in hibernate.
- It is used to map a java.util.Collection property in hibernate.
- It is used to map a java.util.Map property in hibernate.

Q24. Describe the method used to create an HQL Query and SQL Query?

Session.createQuery is used to create a new instance of a query for the HQL query string.

Session.createSQLQuery is used to create a new instance of a query for the SQL query string.

Q25. Explain the important benefits of Hibernate framework?

Few important benefits of Hibernate framework are:

- Hibernate allows us to focus on business logic, eliminating all the boiler-plate code that comes with JDBC and handles the resources.
- Code implementation becomes independent as Hibernate framework provides the support for XML and also to the JPA annotations.
- HQL is powerful Query Language which is similar to SQL, and HQL understands the concepts of polymorphism, inheritance, and association, which makes it fully object-oriented.
- Better performance can be achieved by Hibernate cache.
- It supports Lazy initialization with the use of proxy objects and when required performs actual database queries.
- We can execute native SQL queries using hibernate for vendor specific feature.

On the whole, hibernate makes it a better choice in the current market for ORM tool, as it contains all the features that you will require in an ORM tool.

Q26. Describe the important interfaces of Hibernate framework?

Important interfaces of Hibernate framework are:

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

It is an immutable thread safe cache of compiled mappings for a single database.

We are supposed to initialize SessionFactory once and then we are allowed to cache and reuse it.

The SessionFactory instance is used to return the session objects for database operations.

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

It is a single threaded and short-lived object, which represents a conversation between the persistent store and the application.

The session should be opened only when it is required and should be closed as soon as the user is done.

The session object is the interface between hibernate framework and Java application code and it provides methods for the CRUD operations.

- **Transaction (org.hibernate.transaction)**

It is a single threaded and short-lived object used by the application, which specifies atomic units of work.

The application is abstracted from the underlying JDBC or JTA transaction.

Q27. What is difference between save and persist in hibernate?

Difference between save and persist in Hibernate

Difference	Save	Persist
Return Type	Returns Serializable object	Returns Void

Q28. Explain what is a dialect?

Hibernate Dialect is used to specify the type of database we are going to use. Hibernate requires this to

know in advance so it is able to generate appropriate type of SQL statements based on database type.

Q29. Which annotation is used to declare a class as a hibernate bean ?

@Entity annotation is used to declare a class as an entity.

Example

```
@Entity
@Table(name="posts")
public class Post{
String title;
String description;
}
```

Q30. What inheritance mapping strategies are available in Hibernate?

Hibernate have 3 ways of inheritance mapping, They are

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

Q31. What is transient state in hibernate ?

In Hibernate object support multiple states. Transient state is one of them, an object with the transient state will never be added to the database and it has no persistence representation in hibernate session. A Transient state object is destroyed through garbage collections after an application is closed.

Q32. How to write composite key mapping for hibernate?

The **composite key mapping** in hibernate can be written in two ways, EmbeddedId and IdClass. For a primary key to be composite, it should follow the following rules. The primary key must be public and its properties must be public or protected. The primary key class must be serializable and must define equals and hashCode. The primary key class must also be defined to represent the composite primary key. If the primary key follows these rules, then it can be mapped with composite using EmbeddedId or IdClass.

Q33. How to create two same tables in different Schema's by using single configuration file?

Please Visit OnlineInterviewquestions.com to download more pdfs