

Hibernate Interview Question

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.

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

Hibernate Interview Questions for Fresher

Q1). What is Hibernate?

Hibernate is one of the most extensively used ORM (Object/Relational mapping) tool for Java apps. It's quite widely used a lot in enterprise applications for the purpose of database operations.

Q2). What are the benefits of Hibernate over JDBC?

Other than a good persistence i.e. saving as well as loading the data from a given Database, Hibernate also offer the following aids

- Caching
- Lazy Loading
- The developer is absolutely free from scripting code to either load or store data into the database.
- A good relationship management and offers code for mapping an object right to the data.

Q3). What is HQL?

HQL is the acronym of Hibernate Query Language. It considers the java questions comparably as that of the SQL. It is an Object-Oriented Query Language and is independent of the database.

Q4). Why should you use Hibernate?

A portion of the upsides of Hibernate are:

- It gives Simple Querying of information.
- An application server isn't required to work.
- The complex relationship of objects in the database can be controlled.
- The database gets to is limited with savvy getting procedures.
- It deals with the mapping of Java classes to database tables without composing any code.
- Properties of the XML document is changed if there should arise an occurrence of any required change in the database.

Q5). What are the Key segments of Hibernate?

The Key segments of Hibernate are:

- **Session:** It is utilized to get a physical system with a database.
- **Transaction:** It speaks to the unit of work with a database.
- **Query:** It utilizes SQL and HQL string to recover the information from the database and make objects.
- **Criteria:** It is utilized to make and execute protest situated questions and recover the items.
- **Configuration:** It speaks to the properties of records required by Hibernate
- **Session Factory:** It designs sleep for the application utilizing the gave arrangement document and instantiates the session protest.

Q6). Difference between get() vs load() method in Hibernate?

The key contrast amongst getting () and load() technique is that load() will throw an exception if id passed to them aren't found, however, get() will return null. Another imperative distinction is that load can return intermediary without hitting the database unless required (when you get to any characteristic other than id) yet get() dependably go to the database, so once in a while utilizing load() can be quicker than the get() technique. It bodes well to utilize the load() technique on the off chance that you know the question exists yet to get() strategy if you don't know about object's presence.

Q7). What is the difference between save() and persist() method in Hibernate?

The foremost difference between save() and persist() system is that save returns the result of a Serializable object while the return type of persisting () system is void, so it doesn't yield anything as such.

Q8). What is the difference between the first and second level cache in Hibernate?

This is a likely follow-up question of the previous Hibernate interview question. The first level cache is preserved at the Session level through the second level cache is maintained at a SessionFactory level and is shared by all the sessions. You can consider reading further on caching in Hibernate.

Q9). What is a Session in hibernate?

A Session is something which is used to get a somatic connection with a database. The Session object is lightweight and is planned to be instantiated each time an interaction is needed with the database.

Q10). Has Hibernate Session interface been thread-safe in Java?

The answer is no, Session object is not thread-safe in the Hibernate framework and is proposed to be used within the single thread in the application.

Q11). Shed some light on other ORM frameworks? Is there any substitute for Hibernate?

This is a very common question, occasionally asked to start the conversation on Hibernate and other times to finish the interview. EJB and TopLink frameworks from Oracle are two of the most widely held substitutes of a Hibernate framework.

Q12). What is Transaction in hibernate?

A Transaction signifies a unit of work with the database and maximum of the RDBMS back up the transaction functionalities. Transactions in Hibernate are controlled by a basic operation manager and transaction.

Q13). What is a Query in hibernate?

Query objects use the SQL or Hibernate Query Language (HQL) string to recover data from the database and generate objects. A Query request is used to bind inquiry parameters, bound the number of results given back by the query, and lastly to perform the query.

Q14). What is ORM?

ORM stands for Object/Relational mapping. It is the automatic and transparent persistence of objects present in a Java application into the tables of a relational database by means of the metadata that defines the mapping between the objects and the database.

Q15). For what reason is ORM favored over JDBC?

ORM is favored over JDBC because of the following reasons-

- It permits business access to the articles instead of Database tables.
- It shrouds the subtle elements of SQL questions from OO rationale.
- It depends on JDBC "under the hood".
- Dealing with database execution isn't required.
- Entities depend on business ideas as opposed to database structures.
- It creates the programmed key and Transaction administration.
- Application improvement is quicker.

Q16). What are the different levels of ORM quality?

Four levels defined for ORM quality:

- Pure relational
- Light object mapping
- Medium object mapping
- Full object mapping

Q17). What do you mean by a pure relational ORM?

The whole application, together with the user interface, is planned around the relational model and on SQL-based relational operations.

Q18). What is a SessionFactory in hibernate?

A configuration object is used to make a SessionFactory object which in turn arranges Hibernate for the application by means of the supplied configuration file and then permits for a Session object to be instantiated.

Q19). How do you log SQL queries issued by the Hibernate framework in Java application?

You can procedure the show_sql property to log [SQL queries](#) delivered by the Hibernate framework.

Q20). Highlight the Core interfaces of Hibernate framework?

There are mainly five core interfaces being utilized widely in each Hibernate application. Using these interfaces you can either store or salvage any insistent objects and also regulate transactions.

- Session interface
- Session Factory interface
- Configuration interface
- Transaction interface
- Query and Criteria interfaces

Q21). What is called lazy fetching from Hibernate?

In Hibernate Lazy fetching is concomitant with child objects that are loading for their parent objects. By way of a Hibernate mapping file (.hbm.xml), you can specify the selection of loading the various child objects. By default, Hibernate does not load child objects. Lazy=true command means not to load the child objects.

Q22). Elucidate the difference between merge and update method?

Use update() method only when you are sure that the session does not comprise a previously insistent occurrence with the same identifier, and merge() method in case you want to merge your alterations at any time without deliberation of the state of the session.

Q23). How various components are used in Hibernate?

Components let the object to be stored as a value and not as a reference. It is utilized to be stored directly without the necessity to affirm the properties of the various interfaces or identifiers in the given application. Components in Hibernate describe the empty constructor in which the references that are common are not sustained.

Q24). What are Criteria in hibernate?

Criteria object are utilized to generate and implement [object-oriented](#) criteria queries to recover objects.

Q25). Explain the Transaction object in Hibernate?

It represents a unit of work with the database and the majority of the RDBMS (Relational Database Management System) underpins exchange usefulness. In Hibernate, exchanges are overseen by a basic exchange administrator and exchange from JDBC or JTA. It is a discretionary question and the Hibernate Application don't utilize this interface, rather, they handle the exchanges in their code.

Q26). What is a One-to-One association in Hibernate?

It is like the many-to-one association and the distinction lies in the section that will be set as a unique one. The many-to-one component is utilized to characterize balanced affiliation. To the characterized variable a name property is set in the parent class and the section ascribe is utilized to set segment name in the parent table, or, in other words, that just a single object gets related with another.

Q27). What is One-to-Many association in Hibernate?

In this affiliation, one question can be related to various items. The One-to-Many mapping is actualized utilizing a Set Java accumulation that does not have any excess component. A One-to-Many component of the set component shows the connection of one protest numerous objects.

Q28). What is first level cache in Hibernate?

It is session reserve and compulsory cache. It is from the first level store through which every one of the solicitations must pass. The session question stores an object under its control before submitting it to the database.

Q29). What is the second level cache in Hibernate?

It is a discretionary cache. What's more, dependably the primary level reserve will be counseled before any endeavor is performed to find an object in the second level store. This store can be designed on a pre-accumulation and per-class premise and it is for the most part in charge of reserving objects over the sessions.

Q30). What is Query level cache in Hibernate?.

In sleep, a store question can be actualized that outcomes in sets and incorporates intimately with the second level cache. It is a discretionary component and it requires two extra reserve locales that can hold the stored inquiry results and furthermore the timestamps at whatever point a table is refreshed. This is helpful just for the inquiries that run every now and again holding similar parameters.

1) 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.

2) 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.

3) 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.

4) 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.

5) 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.

6) 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.

7) 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.

8) Explain the Criteria object in Hibernate?

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

9) 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.

10) 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

11) 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.

12) 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.

13) 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.

14) 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.

15) 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.

16) 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.

17) 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.

18) 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.

19) 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.

20) 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.

21) 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.

22) Explain the persistent classes in Hibernate?

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

23) 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.

24) 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.

25) 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.

26) 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.

27) 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

28) 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.

29) 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;
}
```

30) 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

31) 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.

32) 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.

Question Hard Unique :- 10

1. How to complete a transaction in Hibernate?

There are two distinct actions if you're looking to complete a transaction in hibernate:

- Commit
- Rollback

Once a transaction is committed, the transaction data is written to the database. In case there is a rollback, the data exchange is flushed and never written or updated to the database.

2. Can you touch upon the different types of relationships available in Hibernate mapping?

Hibernate consists of three different types of relationships:

- One-to-One mapping
- One-to-Many mapping
- Many-to-Many mapping

3. What is the purpose of Session.beginTransaction()?

Hibernate keeps a log of every data exchange with the help of a transaction. Thereon, in case a new exchange of data is about to get initiated, the function Session.beginTransaction is executed in order to begin the transaction.

4. Where is object/relational mappings defined in Hibernate?

An XML document is where object/relational mappings are usually defined in. This mapping file gives instructions to Hibernate on how to accurately map the defined class/classes to the database tables. In addition, the format .hbm.xml should be used to save the file with the mapping document.

5. What is lazy loading?

Lazy loading is defined as a technique in which objects are loaded on an on-demand basis. It has been enabled by default since the advent of Hibernate 3 to ensure that

child objects are not loaded when the parent is.

6. What are the ways to express joins in HQL?

HQL allows you to express joins in four ways:

- An implicit association join
- A fetch join in the FROM clause
- A theta-style join in the WHERE clause
- An ordinary join in the FROM clause

7. How would you define automatic dirty checking?

Automatic dirty checking can be defined as a feature that helps us in saving the effort of explicitly asking Hibernate to update the database every time we modify or make changes to the state of an object inside a transaction.

8. Explain the different ways Hibernate manages concurrency?

Hibernate has numerous ways of managing concurrency. They are as listed below:

- Automatic versioning
- Detached object
- Extended user sessions

9. Can you detail out the various collection types in Hibernate?

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

- Bag
- Set
- List
- Array
- Map

10. Explain the difference between hibernate and Spring?

While Hibernate is an ORM framework tool used for data persistency, Spring, on the other hand, is a framework for enterprise applications. Moreover, Spring supports Hibernate and provides the different classes which are essentially templates containing the common code.

In addition to the above-mentioned list of questions, here are a few more

Frequently Asked Questions which we have put together to help you crack your all-important Hibernate interviews.

- How many concurrent Hibernate sessions can exist at any point in time in an application?
- Is SessionFactory a thread-safe object?
- Can you explain the role Session interface plays in Hibernate?
- Describe the process of switching between relational databases without code changes.
- Explain the difference between the session.get() method and the session.load() method.
- Is it possible to use Hibernate to map persistent entity POJO to XML files?
- List out all the different states of an instance in Hibernate.
- Can you describe the different contextual sessions in Hibernate?
- Touch upon the pros of Hibernate Criteria API.
- Explain the benefit of Hibernate Tools Eclipse plugin.
- Describe component mapping in Hibernate.