Hibernate Interview questions

**[1) What is Lazy loading in hibernate ?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled21)**

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.

**[2) Explain the persistent classes in Hibernate?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled22)**

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

### ****[3) Explain some of the elements of hbm.xml?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled23)****

* 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.

**[4) Describe the method used to create an HQL Query and SQL Query?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled24)**

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.

### ****[5) Explain the important benefits of Hibernate framework?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled25)****

**Few important benefits of Hibernate framework are:**

* Hibernates 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.

**[6) What is difference between save and persist in hibernate?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled27)**

Difference between save and persist in Hibernate

|  |  |  |
| --- | --- | --- |
| Difference | Save | Persist |
| Return Type | Returns Serializable object | Returns Void |

**[7) Explain what is a dialect?](https://www.onlineinterviewquestions.com/hibernate-interview-questions-answers/" \l "collapseUnfiled28)**

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.

### 8) What is difference between Hibernate Session get() and load() method?

Hibernate session comes with different methods to load data from database. get and load are most used methods, at first look they seems similar but there are some differences between them.

1. get() loads the data as soon as it’s called whereas load() returns a proxy object and loads data only when it’s actually required, so load() is better because it support lazy loading.
2. Since load() throws exception when data is not found, we should use it only when we know data exists.
3. We should use get() when we want to make sure data exists in the database.

### 8) What is hibernate caching? Explain Hibernate first level cache?

As the name suggests, hibernate caches query data to make our application faster. Hibernate Cache can be very useful in gaining fast application performance if used correctly. The idea behind cache is to reduce the number of database queries, hence reducing the throughput time of the application.

Hibernate first level cache is associated with the Session object. Hibernate first level cache is enabled by default and there is no way to disable it. However hibernate provides methods through which we can delete selected objects from the cache or clear the cache completely.  
Any object cached in a session will not be visible to other sessions and

# **Hibernate Lazy Collection**

Lazy collection loads the child objects on demand, it is used to improve performance. Since Hibernate 3.0, lazy collection is enabled by default.

To use lazy collection, you may optionally use lazy="true" attribute in your collection. It is by default true, so you don't need to do this. If you set it to false, all the child objects will be loaded initially which will decrease performance in case of big data.

# **Hibernate Second Level Cache**

**Hibernate second level cache** uses a common cache for all the session object of a session factory. It is useful if you have multiple session objects from a session factory.

**SessionFactory** holds the second level cache data. It is global for all the session objects and not enabled by default.

Different vendors have provided the implementation of Second Level Cache.

1. EH Cache
2. OS Cache
3. Swarm Cache
4. JBoss Cache

Each implementation provides different cache usage functionality. There are four ways to use second level cache.

1. **read-only:** caching will work for read only operation.
2. **nonstrict-read-write:** caching will work for read and write but one at a time.
3. **read-write:** caching will work for read and write, can be used simultaneously.
4. **transactional:** caching will work for transaction.

The cache-usage property can be applied to class or collection level in hbm.xml file. The example to define cache usage is given below

# What is Session object in Hibernate?

Unlike SessionFactory, the Session object will be created on demand. Session is a lightweight object. Session provides a physical connectivity between your application and database. The Session will be established each time your application wants do something with database. Session object will be provided by SessionFactory object. All the persistent objects will be saved and retrieved through Session object. The session object must be destroyed after using it.

The lifecycle of a Session is bounded by the beginning and end of a logical transaction. The main function of the Session is to offer create, read and delete operations for instances of mapped entity classes. Instances may exist in one of three states:

* ***transient***: never persistent, not associated with any Session.
* ***persistent***: associated with a unique Session.
* ***detached***: previously persistent, not associated with any Session.

Here is the sample implementation for getting Session object from SessionFactory: