Hibernate - Interview Questions

Dear readers, these **Hibernate Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **Hibernate**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer −

What is JDBC?

JDBC stands for Java Database Connectivity and provides a set of Java API for accessing the relational databases from Java program. These Java APIs enables Java programs to execute SQL statements and interact with any SQL compliant database.

What is ORM?

ORM stands for **O**bject-**R**elational **M**apping (ORM) is a programming technique for converting data between relational databases and object oriented programming languages such as Java, C# etc.

What are the advantages of ORM over JDBC?

An ORM system has following advantages over plain JDBC

Sr.No.	Advantages
1	Lets business code access objects rather than DB tables.
2	Hides details of SQL queries from OO logic.
3	Based on JDBC 'under the hood'
4	No need to deal with the database implementation.
5	Entities based on business concepts rather than database structure.
6	Transaction management and automatic key generation.
7	Fast development of application.

Name some of the ORM frameworks based on JAVA.

There are several persistent frameworks and ORM options in Java.

- Enterprise JavaBeans Entity Beans
- Java Data Objects
- Castor
- TopLink
- Spring DAO
- Hibernate

What is Hibernate?

Hibernate is an Object-Relational Mapping(ORM) solution for JAVA and it raised as an open source persistent framework created by Gavin King in 2001. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieve the developer from 95% of common data persistence related programming tasks.

What are the advantages of using Hibernate?

Following are the advantages of using Hibernate.

- Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
- Provides simple APIs for storing and retrieving Java objects directly to and from the database.
- If there is change in Database or in any table then the only need to change XML file properties.
- Abstract away the unfamiliar SQL types and provide us to work around familiar Java Objects.
- Hibernate does not require an application server to operate.
- Manipulates Complex associations of objects of your database.
- Minimize database access with smart fetching strategies.
- Provides Simple querying of data.

Name some of the databases that hibernate supports.

Hibernate supports almost all the major RDBMS. Following is list of few of the database engines supported by Hibernate.

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

Name some of the java based tools/frameworks that supports hibernate integration.

Hibernate supports a variety of other technologies, including the following −

- XDoclet Spring
- J2EE
- Eclipse plug-ins
- Maven

What are the key components/objects of hibernate?

Following are the key components/objects of Hibernate −

- **Configuration** − Represents a configuration or properties file required by the Hibernate.
- **SessionFactory** − Configures Hibernate for the application using the supplied configuration file and allows for a Session object to be instantiated.
- **Session** − Used to get a physical connection with a database.
- **Transaction** − Represents a unit of work with the database and most of the RDBMS supports transaction functionality.
- **Query** − Uses SQL or Hibernate Query Language (HQL) string to retrieve data from the database and create objects.
- **Criteria** − Used to create and execute object oriented criteria queries to retrieve objects.

What are the two key components of a hibernate configuration object?

The Configuration object provides two keys components −

- **Database Connection** − This is handled through one or more configuration files supported by Hibernate. These files are **hibernate.properties** and **hibernate.cfg.xml**.
- **Class Mapping Setup**

This component creates the connection between the Java classes and database tables.

What is a configuration object in hibernate?

The Configuration object is the first Hibernate object you create in any Hibernate application and usually created only once during application initialization. It represents a configuration or properties file required by the Hibernate.

What is a SessionFactory in hibernate?

Configuration object is used to create a SessionFactory object which inturn configures Hibernate for the application using the supplied configuration file and allows for a Session object to be instantiated. The SessionFactory is a thread safe object and used by all the threads of an application.

The SessionFactory is heavyweight object so usually it is created during application start up and kept for later use. You would need one SessionFactory object per database using a separate configuration file. So if you are using multiple databases then you would have to create multiple SessionFactory objects.

What is Session in hibernate?

A Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database. Persistent objects are saved and retrieved through a Session object.

The session objects should not be kept open for a long time because they are not usually thread safe and they should be created and destroyed them as needed.

What is Transaction in hibernate?

A Transaction represents a unit of work with the database and most of the RDBMS supports transaction functionality. Transactions in Hibernate are handled by an underlying transaction manager and transaction (from JDBC or JTA).

This is an optional object and Hibernate applications may choose not to use this interface, instead managing transactions in their own application code.

What is Query in hibernate?

Query objects use SQL or Hibernate Query Language (HQL) string to retrieve data from the database and create objects. A Query instance is used to bind query parameters, limit the number of results returned by the query, and finally to execute the query.

What is Criteria in hibernate?

Criteria object are used to create and execute object oriented criteria queries to retrieve objects.

Name some of the properties you would require to configure for a databases in a standalone situation.

Sr.No.	Properties & Description
1	**hibernate.dialect** This property makes Hibernate generate the appropriate SQL for the chosen database.
2	**hibernate.connection.driver_class** The JDBC driver class.
3	**hibernate.connection.url** The JDBC URL to the database instance.
4	**hibernate.connection.username** The database username.
5	**hibernate.connection.password** The database password.
6	**hibernate.connection.pool_size** Limits the number of connections waiting in the Hibernate database connection pool.
7	**hibernate.connection.autocommit** Allows autocommit mode to be used for the JDBC connection.

What are the three states of a persistent entity at a given point in time?

Instances may exist in one of the following three states at a given point in time −

- **transient** − A new instance of a a persistent class which is not associated with a Session and has no representation in the database and no identifier value is considered transient by Hibernate.
- **persistent** − You can make a transient instance persistent by associating it with a Session. A persistent instance has a representation in the database, an identifier value and is associated with a Session.
- **detached** − Once we close the Hibernate Session, the persistent instance will become a detached instance.

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

Session.beginTransaction method begins a unit of work and returns the associated Transaction object.

Which method is used to add a criteria to a query?

Session.createCriteria creates a new Criteria instance, for the given entity class, or a superclass of an entity class.

Which method is used to create a HQL query?

Session.createQuery creates a new instance of Query for the given HQL query string.

Which method is used to create a SQL query?

Session.createSQLQuery creates a new instance of SQLQuery for the given SQL query string.

Which method is used to remove a persistent instance from the datastore?

Session.delete removes a persistent instance from the datastore.

Which method is used to get a persistent instance from the datastore?

Session.get returns the persistent instance of the given named entity with the given identifier, or null if there is no such persistent instance.

Which method is used to re-read the state of the given instance from the underlying database?

Session.refresh re-reads the state of the given instance from the underlying database.

Which method is used to save the state of the given instance from the underlying database?

Session.save saves the state of the given instance from the underlying database.

Which method is used to update the state of the given instance from the underlying database?

Session.update updates the state of the given instance from the underlying database.

Which method is used to save or update the state of the given instance from the underlying database?

Session.saveOrUpdate either saves(Object) or updates(Object) the given instance.

What are persistent classes in hibernate?

Java classes whose objects or instances will be stored in database tables are called persistent classes in Hibernate.

What are the best practices that hibernate recommends for persistent classes.

There are following main rules of persistent classes, however, none of these rules are hard requirements.

- All Java classes that will be persisted need a default constructor.
- All classes should contain an ID in order to allow easy identification of your objects within Hibernate and the database. This property maps to the primary key column of a database table.
- All attributes that will be persisted should be declared private and have **getXXX** and **setXXX** methods defined in the JavaBean style.
- A central feature of Hibernate, proxies, depends upon the persistent class being either non-final, or the implementation of an interface that declares all public methods.
- All classes that do not extend or implement some specialized classes and interfaces required by the EJB framework.

Where Object/relational mappings are defined in hibernate?

An Object/relational mappings are usually defined in an XML document. This mapping file instructs Hibernate how to map the defined class or classes to the database tables. We should save the mapping document in a file with the format <classname>.hbm.xml.

What is root node of hbm.xml?

The mapping document is an XML document having **<hibernate-mapping>** as the root element which contains all the <class> elements.

Which element of hbm.xml defines a specific mappings from a Java classes to the database tables?

The **<class>** elements are used to define specific mappings from a Java classes to the database tables. The Java class name is specified using the **name** attribute of the class element and the database table name is specified using the **table** attribute.

Which element of hbm.xml defines maps the unique ID attribute in class to the primary key of the database table?

The **<id>** element maps the unique ID attribute in class to the primary key of the database table. The **name** attribute of the id element refers to the property in the class and the **column** attribute refers to the column in the database table. The **type** attribute holds the hibernate mapping type, this mapping types will convert from Java to SQL data type.

Which element of hbm.xml is used to automatically generate the primary key values?

The **<generator>** element within the id element is used to automatically generate the primary key values. Set the **class** attribute of the generator element is set to **native** to let hibernate pick up either **identity, sequence** or **hilo** algorithm to create primary key depending upon the capabilities of the underlying database.

Which element of hbm.xml is used to map a Java class property to a column in the database table?

The **<property>** element is used to map a Java class property to a column in the database table. The **name** attribute of the element refers to the property in the class and the **column** attribute refers to the column in the database table. The **type** attribute holds the hibernate mapping type, this mapping types will convert from Java to SQL data type.

Which element of hbm.xml is used to map a java.util.Set property in hibernate?

This is mapped with a <set> element and initialized with java.util.HashSet.

Which element of hbm.xml is used to map a java.util.SortedSet property in hibernate?

This is mapped with a <set> element and initialized with java.util.TreeSet. The sort attribute can be set to either a comparator or natural ordering.

Which element of hbm.xml is used to map a java.util.List property in hibernate?

This is mapped with a <list> element and initialized with java.util.ArrayList.

Which element of hbm.xml is used to map a java.util.Collection property in hibernate?

This is mapped with a <bag> or <ibag> element and initialized with java.util.ArrayList.

Which element of hbm.xml is used to map a java.util.Map property in hibernate?

This is mapped with a <map> element and initialized with java.util.HashMap.

Which element of hbm.xml is used to map a java.util.SortedMap property in hibernate?

This is mapped with a <map> element and initialized with java.util.TreeMap. The sort attribute can be set to either a comparator or natural ordering.

What is many-to-one association?

A many-to-one association is the most common kind of association where an Object can be associated with multiple objects. For example a same address object can be associated with multiple employee objects.

<many-to-one> element is used to define many-to-one association. The name attribute is set to the defined variable in the parent class. The column attribute is used to set the column name in the parent table.

What is one-to-one association?

A one-to-one association is similar to many-to-one association with a difference that the column will be set as unique. For example an address object can be associated with a single employee object.

<many-to-one> element is used to define one-to-one association. The name attribute is set to the defined variable in the parent class. The column attribute is used to set the column name in the parent table which is set to unique so that only one object can be associated with an other object.

What is one-to-many association?

In One-to-Many mapping association, an object can be can be associated with multiple objects. For example Employee object relates to many Certificate objects.

A One-to-Many mapping can be implemented using a Set java collection that does not contain any duplicate element.

<one-to-many> element of set element indicates that one object relates to many other objects.

What is many-to-many association?

A Many-to-Many mapping can be implemented using a Set java collection that does not contain any duplicate element.

<many-to-many> element indicates that one object relates to many other objects and column attributes are used to link intermediate column.

Is SessionFactory a thread-safe object?

Yes, SessionFactory is a thread-safe and can be accessed by multiple threads simultaneously.

Is Session a thread-safe object?

No, Session is not thread-safe.

What is the difference between save() and persist() methods of session object?

session.save saves the object and returns the id of the instance whereas persist do not return anything after saving the instance.

What is the difference between get() and load() methods of session object?

There are following differences between get() and load() methods.

- get() returns null if no data is present where as load throws ObjectNotFoundException exception in such case.

- get() always hits the database whereas load() method doesn't hit the database.
- get() returns actual object whereas load() returns proxy object.
- A central feature of Hibernate, proxies, depends upon the persistent class being either non-final, or the implementation of an interface that declares all public methods.
- All classes that do not extend or implement some specialized classes and interfaces required by the EJB framework.

What is lazy loading?

Lazy loading is a technique in which objects are loaded on demand basis. Since Hibernate 3, lazy loading is by default, enabled so that child objects are not loaded when parent is loaded.

What is HQL?

HQL stands for Hibernate Query Language. It takes java objects in the same way as SQL takes tables. HQL is a Object Oriented Query language and is database independent.

What is first level cache in hibernate?

The first-level cache is the Session cache and is a mandatory cache through which all requests must pass. The Session object keeps an object under its own power before committing it to the database.

What is second level cache in hibernate?

What is Query level cache in hibernate?

Hibernate also implements a cache for query resultsets that integrates closely with the second-level cache.

This is an optional feature and requires two additional physical cache regions that hold the cached query results and the timestamps when a table was last updated. This is only useful for queries that are run frequently with the same parameters.

What are concurrency strategies?

A concurrency strategy is a mediator which responsible for storing items of data in the cache and retrieving them from the cache. If you are going to enable a second-level cache, you will have to decide, for each persistent class and collection, which cache concurrency strategy to use.

- **Transactional** − Use this strategy for read-mostly data where it is critical to prevent stale data in concurrent transactions,in the rare case of an update.

- **Read-write** − Again use this strategy for read-mostly data where it is critical to prevent stale data in concurrent transactions,in the rare case of an update.
- **Nonstrict-read-write** − This strategy makes no guarantee of consistency between the cache and the database. Use this strategy if data hardly ever changes and a small likelihood of stale data is not of critical concern.
- **Read-only** − A concurrency strategy suitable for data which never changes. Use it for reference data only.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)