JAVA means DURGA SOFT

JAVA FAQ`S

Very important Questions in HIBERNATE

DURGA M.Tech

(Sun certified & Realtime Expert)

Ex. IBM Employee

Trained Lakhs of Students
for last 14 years across INDIA

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

HIBERNATE FAQs

1.what is the advantage of Hibernate over jdbc?

There are so many
1) Hibernate is data base independent, your code will work for all ORACLE, MySQL ,SQL Server etc. In case of JDBC query must be data base specific. So hibernate based persistance logic is database independent persistence logic and JDBC based persistence logic is database dependent logic.
2) As Hibernate is set of Objects ,
3) No need to learn SQL language. You can treat TABLE as a Object . Only Java knowledge is need. In case of JDBC you need to learn SQL.
3) Don't need Query tuning in case of Hibernate. If you use Criteria Quires in Hibernate then hibernate automatically tuned your query and return best result with performance.
In case of JDBC you need to tune your queries.
4) You will get benefit of Cache. Hibernate support two level of cache. First level and 2nd level. So you can store your data into Cache for better performance.
In case of JDBC you need to implement your java cache .
5) Hibernate supports Query cache and It will provide the statistics about your query and database status.
JDBC Not provides any statistics.
6) Development fast in case of Hibernate because you don't need to write queries
7) No need to create any connection pool in case of Hibernate. You can use c3p0.
In case of JDBC you need to write your own connection pool
8) In the xml file you can see all the relations between tables in case of Hibernate. Easy readability.
9) You can load your objects on start up using lazy=false in case of Hibernate.
JDBC Dont have such support.
10) Hibernate Supports automatic versioning of rows but JDBC Not.

2.What is Hibernate?
Hibernate is an open source, light weight Object Relational Mapping tool to develop the database independent persistence login in java and j2ee based applications.
Hibernate is a pure Java object-relational mapping (ORM) and persistence framework that allows you to map plain old Java objects to relational database tables using (XML) configuration and mapping files. Its purpose is to relieve the developer from a significant amount of relational data persistence-related programming tasks
3.What is ORM ?
ORM stands for object/relational mapping, means providing the mapping between class with table and member variables with columns is called ORM. ORM is the automated persistence of objects in a Java application to the tables in a relational database.

4.hat does ORM consists of ?
An ORM solution consists of the following four pieces:

- API for performing basic CRUD operations
- API to express queries referring to classes
- Facilities to specify metadata
- Optimization facilities : dirty checking,lazy associations fetching

5.What are the ORM levels ?
The ORM levels are:

- Pure relational (stored procedure.)
- Light objects mapping (JDBC)
- Medium object mapping
- Full object Mapping (composition, inheritance, polymorphism, persistence by reachability)

.
6.Why do you need ORM tools like hibernate?
The main advantage of ORM like hibernate is that it can develop the database independent persistence logic. Apart from this, ORM provides following benefits:

- **Improved productivity**
  - High-level object-oriented API
  - Less Java code to write
  - No SQL to write
- **Improved performance**
  - Sophisticated caching
  - Lazy loading
  - Eager loading
- **Improved maintainability**
  - A lot less code to write
- **Improved portability**
- ORM framework generates database-specific SQL for you

7.What Does Hibernate Simplify?
Hibernate simplifies:

- Saving and retrieving your domain objects
- Making database column and table name changes
- Centralizing pre save and post retrieve logic
- Complex joins for retrieving related items
- Schema creation from object model

8.What is the main difference between Entity Beans and Hibernate ?

1)In Entity Bean at a time we can interact with only one data Base. Where as in Hibernate we can able to establishes the connections to more than One Data Base. Only thing we need to write one more configuration file.

2) EJB need container like Weblogic, WebSphare but hibernate don't nned. It can be run on tomcat.

3) Entity Beans does not support OOPS concepts where as Hibernate does.

4) Hibernate supports multi level cacheing, where as Entity Beans doesn't.

5) In Hibernate C3P0 can be used as a connection pool.

6) Hibernate is container independent. EJB not.

9.What are the Core interfaces and classes of Hibernate framework?
The five core interfaces are used in just about every Hibernate application. Using these interfaces, you can store and retrieve persistent objects and control transactions.

- Configuration class (org.hibernate.cfg package)
- Session interface (org.hibernate package)
- SessionFactory interface (org.hibernate package)
- Transaction interface (org.hibernate package)
- Query and Criteria interfaces (org.hibernate package)

10.What is the general flow of Hibernate communication with RDBMS?
The general flow of Hibernate communication with RDBMS is :

- Load the Hibernate configuration file and create configuration object. It will automatically load all hbm mapping files because mapping file can be configured in configuration file.
- Create session factory from configuration object
- Get one session from this session factory
- Create HQL Query
- Execute query to get list containing Java objects.

11.What is the need for Hibernate mapping file?
Hibernate mapping file is used to provides the mapping between java class with table member variables with column names of the table. And also we can configure primary key generation algorithm, relations and so on. Typical mapping file look as follows:

12.What are the important tags of hibernate.cfg.xml?

This file can be used to provide the database information like driverclass name, url, database usename, database password, dialect, connection pooling mapping file and so on.
Following are the important tags of hibernate.cfg.xml:

13.What role does the Session interface play in Hibernate?
The main runtime interface between a Java application and Hibernate The Session interface is the primary interface used by Hibernate applications. It is a single-threaded, short-lived object representing a conversation between the application and the persistent store. It allows you to create query objects to retrieve persistent objects.
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

Session session = sessionFactory.openSession();
Session interface role:

- Wraps a JDBC connection
- Factory for Transaction
- Holds a mandatory (first-level) cache of persistent objects, used when navigating the object graph or looking up objects by identifier

14.What role does the SessionFactory interface play in Hibernate?
SessionFactorys are immutable. The behaviour of a SessionFactory is controlled by properties supplied at configuration time. These properties are defined on Environment.
The application obtains Session instances from a SessionFactory. There is typically a single SessionFactory for the whole application—created during application initialization. The SessionFactory caches generate SQL statements and other mapping metadata that Hibernate uses at runtime. It also holds cached data that has been read in one unit of work and may be reused in a future unit of work
Implementors must be threadsafe.
SessionFactory sessionFactory = configuration.buildSessionFactory();

15.What are the most common ways to specify the Hibernate configuration properties?
The most common methods of Hibernate configuration are:

- Programmatic configuration

By using setProperty(-) method of org.hibernate.cfg.Configuration.

- XML configuration (hibernate.cfg.xml)
- By using .properties file
- By Using annotaions.(from Hibernate 3.3 on words)

16.How do you map Java Objects with Database tables?

- First we need to write Java domain objects (beans with setter and getter).
- Write hbm.xml, where we map java class to table and database columns to Java class variables.

Example :

```
<hibernate-mapping>
  <class name="com.durgasoft.EmployeeBean"  table="EMPLOYEE">
    <id name=”eid” colume=”id”/>
   <property name="ename" column="NAME" length="255"
    not-null="true"  type="java.lang.String"/>
   <property name="address" column="ADDR" length="255"
    not-null="true"  type="java.lang.String"/>
 </class>
</hibernate-mapping>
```

17.How do you define sequence generated primary key algorithm in hibernate?
By using <id>, <generator> tags we can configure the primary key and primary key generation algorithm.
Example:-

```
<id name="userid" column="USER_ID" type="java.lang.Long">
  <generator class="sequence">
    <param name="table">SEQ_NAME</param>
  <generator>
</id>
```

18.What is component mapping in Hibernate?

- A component is an object saved as a value, not as a reference
- A component can be saved directly without needing to declare interfaces or identifier properties
- Required to define an empty constructor
- Shared references not supported

19 . Difference between getCurrentSession() and openSession() in Hibernate ?

getCurrentSession() :
Obtains the current session. The "current session" refers to a Hibernate Session bound by Hibernate behind the scenes, to the transaction scope.
A Session is opened when getCurrentSession() is called for the first time and closed when the transaction ends. It is also flushed automatically before the transaction commits. You can call getCurrentSession() as often and anywhere you want as long as the transaction runs. Only the Session that you obtained with sf.getCurrentSession() is flushed and closed automatically.

openSession() :
If you decide to use manage the Session yourself the go for sf.openSession() , you have to flush() and close() it.
It does not flush and close() automatically.
Example :

```
Transaction tx =session.berginTransaction();

Session session = factory.openSession();

try {
tx.begin();

// Do some work
session.createQuery(...);
session.persist(...);

session.flush(); // Extra work you need to do

tx.commit();
}
catch (RuntimeException e) {
tx.rollback();
throw e; // or display error message
}
finally {
session.close(); // Extra work you need to do
}
```

20.What are the types of Hibernate instance states ?
Three types of instance states:

- Transient -The instance is not associated with any persistence context
- Persistent -The instance is associated with a persistence context

- Detached -The instance was associated with a persistence context which has been closed – currently not associated

21.What are the types of inheritance models in Hibernate?
There are three types of inheritance models in Hibernate:

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

22.What is Hibernate Query Language (HQL)?
Hibernate Query Language is query language which is used to develop the data independent query language in the application. This HQL queries are not related to any database. Hibernate offers a query language that embodies a very powerful and flexible mechanism to query, store, update, and retrieve objects from a database. This language, the Hibernate query Language (HQL), is an object-oriented extension to SQL.

23.What are the ways to express joins in HQL?
HQL provides four ways of expressing (inner and outer) joins:-

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

24 . Transaction with plain JDBC in Hibernate ?

If you don't have JTA and don't want to deploy it along with your application, you will usually have to fall back to JDBC transaction demarcation. Instead of calling the JDBC API you better use Hibernate's Transaction and the built-in session-per-request functionality:

To enable the thread-bound strategy in your Hibernate configuration:

set hibernate.transaction.factory_class to org.hibernate.transaction.JDBCTransactionFactory
set hibernate.current_session_context_class to thread

```
Session session = factory.openSession();
Transaction tx = null;
try {
tx = session.beginTransaction();

// Do some work
session.load(...);
session.persist(...);

tx.commit(); // Flush happens automatically
}
catch (RuntimeException e) {
tx.rollback();
throw e; // or display error message
}
```

```
finally {
session.close();
}
```

25 . What are the general considerations or best practices for defining your Hibernate persistent classes?

1.You must have a default no-argument constructor for your persistent classes and there should be getXXX()and setXXX() methods for all your persistable instance variables.

2.You should implement the equals() and hashCode() methods based on your business key and it is important not to use the id field in your equals() and hashCode() definition if the id field is a surrogate key (i.e. Hibernate managed identifier). This is because the Hibernate only generates and sets the field when saving the object.

3. It is recommended to implement the Serializable interface. This is potentially useful if you want to migrate around a multi-processor cluster.

4.The persistent class should not be final because if it is final then lazy loading cannot be used by creating proxy objects.

26 . Difference between session.update() and session.lock() in Hibernate ?

The session.update method is used to update the persistence object in the in the database.
The session.lock() method simply reattaches the object to the session without checking or updating the database on the assumption that the database in sync with the detached object. It is the best practice to use either session.update(..) or session.saveOrUpdate(). Use session.lock() only if you are absolutely sure that the detached object is in sync with your detached object or if it does not matter because you will be overwriting all the columns that would have changed later on within the same transaction.

27.What are the Collection types in Hibernate ?

- Set
- List
- Array
- Map
- Bag

28.What is the difference between sorted and ordered collection in hibernate?
sorted collection vs. order collection :-

sorted collection	order collection
A sorted collection is sorting a collection by utilizing the sorting features provided by the Java collections framework. The sorting occurs in the memory of JVM which running Hibernate, after the data being read from database using java comparator.	Order collection is sorting a collection by specifying the order-by clause for sorting this collection when retrieval.
If your collection is not large, it will be more efficient way to sort it.	If your collection is very large, it will be more efficient way to sort it .

29.What are the ways to express joins in HQL?
HQL provides four ways of expressing (inner and outer) joins:-

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

30.What do you mean by Named – SQL query?
Named SQL queries are defined in the mapping xml document and called wherever required.
Example:

```
<sql-query name = "empdetails">
  <return alias="emp" class="com.durgasoft.Employee"/>
    SELECT emp.EMP_ID AS {emp.empid},
           emp.EMP_ADDRESS AS {emp.address},
           emp.EMP_NAME AS {emp.name}
    FROM Employee EMP WHERE emp.NAME LIKE :name
</sql-query>
```

Invoke Named Query :

```
List people = session.getNamedQuery("empdetails")
                .setString("TomBrady", name)
                .setMaxResults(50)
                .list();
```

31.How do you invoke Stored Procedures?

```
<sql-query name="selectAllEmployees_SP" callable="true">
 <return alias="emp" class="employee">
  <return-property name="empid" column="EMP_ID"/>
  <return-property name="name" column="EMP_NAME"/>
  <return-property name="address" column="EMP_ADDRESS"/>
   { ? = call selectAllEmployees() }
 </return>
</sql-query>
```

32.Explain Criteria API

The interface org.hibernate.Criteria represents a query against a particular persistent class. The Session is a factory for Criteria instances. Criteria is a simplified API for retrieving entities by composing Criterion objects. This is a very convenient approach for functionality like "search" screens where there is a variable number of conditions to be placed upon the result set.

Example :

```
List employees = session.createCriteria(Employee.class)
                    .add(Restrictions.like("name", "a%") )
                    .add(Restrictions.like("address", "Boston"))
                    .addOrder(Order.asc("name") )
                    .list();
```

33.What's the difference between load() and get()?

load()	get()
Only use the load() method if you are sure that the object exists.	If you are not sure that the object exists, then use one of the get() methods.
load() method will throw an exception if the unique id is not found in the database.	get() method will return null if the unique id is not found in the database.
load() just returns a proxy by default and database won't be hit until the proxy is first invoked.	get() will hit the database immediately.

34.What is the difference between and merge and update ?

Use update() if you are sure that the session does not contain an already persistent instance with the same identifier, and merge() if you want to merge your modifications at any time without consideration of the state of the session.

35.Define cascade and inverse option in one-many mapping?

cascade - enable operations to cascade to child entities.
cascade="all|none|save-update|delete|all-delete-orphan"

inverse - mark this collection as the "inverse" end of a bidirectional association.
inverse="true|false"
Essentially "inverse" indicates which end of a relationship should be ignored, so when persisting a

parent who has a collection of children, should you ask the parent for its list of children, or ask the children who the parents are?

36.Define HibernateTemplate?
org.springframework.orm.hibernate.HibernateTemplate is a helper class which provides different methods for querying/retrieving data from the database. It also converts checked HibernateExceptions into unchecked DataAccessExceptions.

37.What are the benefits does HibernateTemplate provide?
The benefits of HibernateTemplate are :

- HibernateTemplate, a Spring Template class simplifies interactions with Hibernate Session.
- Common functions are simplified to single method calls.
- Sessions are automatically closed.
- Exceptions are automatically caught and converted to runtime exceptions.

38. How do you switch between relational databases without code changes?
Using Hibernate SQL Dialects , we can switch databases. Hibernate will generate appropriate hql queries based on the dialect defined.

39.If you want to see the Hibernate generated SQL statements on console, what should we do?
By using “show_sql” property of the hibernate configuration file
In Hibernate configuration file set as follows:
<property name="show_sql">true</property>

40.What are derived properties?
The properties that are not mapped to a column, but calculated at runtime by evaluation of an expression are called derived properties. The expression can be defined using the formula attribute of the element.

41.Define cascade and inverse option in one-many mapping?
cascade - enable operations to cascade to child entities.
cascade="all|none|save-update|delete|all-delete-orphan"

inverse - mark this collection as the "inverse" end of a bidirectional association.
inverse="true|false"
Essentially "inverse" indicates which end of a relationship should be ignored, so when persisting a parent who has a collection of children, should you ask the parent for its list of children, or ask the children who the parents are?

42 . Explain about transaction file?

Transactions denote a work file which can save changes made or revert back the changes. A transaction can be started by session.beginTransaction() and it uses JDBC connection, CORBA or JTA. When this session starts several transactions may occur.

43 . Difference between session.save() , session.saveOrUpdate() and session.persist()?

All methods are used to store the data in to database

session.save() : save() method uSave does an insert and will fail if the primary key is already persistent.

session.saveOrUpdate() : saveOrUpdate() insert the data in the database if that primary key data not available and it update the data if primary key data not availabt

session.persist() :it is the same like session.save(). But session.save() return Serializable object but session.persist() return void.

For Example :

if you do :-

System.out.println(session.save(question));

This will print the generated primary key.

if you do :-

System.out.println(session.persist(question));

Compile time error because session.persist() return void.

44 . Explain about the id field?

This id field is used to configure the primary key in the mapping file, and also we can configure primary key generation algorithm.

45.What is the use of dynamic-insert and dynamic-update attributes in a class mapping?

Criteria is a simplified API for retrieving entities by composing Criterion objects. This is a very convenient approach for functionality like "search" screens where there is a variable number of conditions to be placed upon the result set.

- dynamic-update (defaults to false): Specifies that UPDATE SQL should be generated at runtime and contain only those columns whose values have changed
- dynamic-insert (defaults to false): Specifies that INSERT SQL should be generated at runtime and contain only the columns whose values are not null.

46.What is automatic dirty checking?

Automatic dirty checking is a feature that saves us the effort of explicitly asking Hibernate to update the database when we modify the state of an object inside a transaction.

47.What are Callback interfaces?
Callback interfaces allow the application to receive a notification when something interesting happens to an object—for example, when an object is loaded, saved, or deleted. Hibernate applications don't need to implement these callbacks, but they're useful for implementing certain kinds of generic functionality.

48.What is Hibernate proxy?
The proxy attribute enables lazy initialization of persistent instances of the class. Hibernate will initially return CGLIB proxies which implement the named interface. The actual persistent object will be loaded when a method of the proxy is invoked.

49.How can Hibernate be configured to access an instance variable directly and not through a setter method ?
By mapping the property with access="field" in Hibernate metadata. This forces hibernate to bypass the setter method and access the instance variable directly while initializing a newly loaded object.

50.How can a whole class be mapped as immutable?
Mark the class as mutable="false" (Default is true),. This specifies that instances of the class are (not) mutable. Immutable classes, may not be updated or deleted by the application.

51 . Explain about transparent persistence of Hibernate?
Transparent persistence is provided for Plain old Java objects or POJOs. For proper functioning of the applications importance should be given to the methods equals () and hash Code methods (). It has a requirement which should be strictly followed in the applications which is a no-argument constructor.

52 . Explain about the dirty checking feature of Hibernate?
Dirty checking feature of the Hibernate allows users or developers to avoid time consuming data base write actions. This feature makes necessary updations and changes to the fields which require a change, remaining fields are left unchanged or untouched.

53 . What is the effect when a transient mapped object is passed onto a Sessions save?
When a Sessions save () is passed to a transient mapped object it makes the method to become more persistent. Garbage collection and termination of the Java virtual machine stays as long as it is deleted explicitly. It may head back to its transient state.

54 . Explain about add Class function?
This function translates a Java class name into file name. This translated file name is then loaded as an input stream from the Java class loader. This add Class function is important if you want efficient usage of classes in your code.